UNIVERSITI TEKNOLOGI MARA

IMAGE DETECTION FOR MONITORING SOFT SHELL CRAB USING INTERNET OF THINGS

MOHAMMAD AFFAIQ BIN AINI

MSc

June 2026

UNIVERSITI TEKNOLOGI MARA

IMAGE DETECTION FOR MONITORING SOFT SHELL CRAB USING INTERNET OF THINGS

MOHAMMAD AFFAIQ BIN AINI

Thesis submitted in fulfilment
of the requirements for the degree of
Master of Science
(Information Technology)

Faculty of Computer and Mathematical Sciences

June 2026

CONFIRMATION BY PANEL OF EXAMINERS

I certify that a Panel of Examiners has met on 19th January 2026 to conduct the final examination of Mohammad Affaiq Bin Aini on his Master of Science thesis entitled "Image Detection for Monitoring Soft Shell Crab Using Internet of Things" in accordance with Universiti Teknologi MARA Act 1976 (Akta 173). The Panel of Examiners recommends that the student be awarded the relevant degree. The Panel of Examiners was as follows:

Nor Azizah M. Yacob, PhD
Associate Professor
Faculty Of Computer And Mathematical Sciences
Universiti Teknologi MARA
(Chairperson)

Nor Ashikin Mohamad Kamal, PhD
Senior Lecturer
Faculty Of Computer And Mathematical Sciences
Universiti Teknologi MARA
(Internal Examiner)

Wang Yin Chai, PhD
Professor
Faculty of Computer Science & Information Technology
Universiti Malaysia Sarawak
(External Examiner)

PROFESSOR DR HJH ZURAEDA IBRAHIM
Dean
Institute of Postgraduate Studies
Universiti Teknologi MARA
Date: 9 June 2026

AUTHOR'S DECLARATION

I declare that the work in this thesis was carried out in accordance with the regulations of Universiti Teknologi MARA. It is original and is the results of my own work, unless otherwise indicated or acknowledged as referenced work. This thesis has not been submitted to any other academic institution or non-academic institution for any degree or qualification.

I, hereby, acknowledge that I have been supplied with the Academic Rules and Regulations for Postgraduate, Universiti Teknologi MARA, regulating the conduct of my study and research.

Name of Student	:	Mohammad Affaiq Bin Aini
Student ID. No.	:	2022803936
Programme	:	Master of Science (Information Technology) – CDCS751
Faculty	:	Computer and Mathematical Sciences
Thesis Title	:	Image Detection for Monitoring Soft Shell Crab Using Internet of Things
Signature of Student	:	…………………………………………
Date	:	June 2026

ABSTRACT

This study addresses the critical challenge of accurately detecting soft-shell crab in an open greenhouse aquaculture environment, where image distortion, glare, and reflections hinder reliable monitoring. The focus is on enhancing the quality of images for crab detection through advanced image rectification and enhancement techniques, integrated with the YOLOv7 object detection model. Crab detection is the first step of soft-shell crab detection. In soft-shell crab farming, the crab detection is labour-intensive, requiring farmers to monitor crabs every two to three hours. By leveraging computer vision and deep learning technologies, this study aims to reduce manual labour and increase productivity. To tackle wide-angle lens distortions, Fisheye Image Correction and 2D-Image Flat-Field Correction are employed, reducing the uneven lighting and spatial distortion caused by the greenhouse setup. While these methods greatly enhance image clarity, significant challenges remain in mitigating reflections and glare that occur in the highly reflective greenhouse environment. Reflection reduction is achieved using the Non-Local Means (NLM) filter and 2D-Image Flat-Field Correction. The methodology involves capturing images using an IoT-based setup, where cameras mounted on rails monitor crabs in containers with 300 images taken. To address the limited dataset, data augmentation techniques are applied, which improve the training of the YOLOv7 model for more accurate detection. Experimental results show that data augmentation boosts detection performance by 7%, and the combination of Fisheyes Image Correction and Flat-Field Correction yields an overall detection precision of 90.3%, compared to 55.4% without image processing. This study demonstrates the effectiveness of the proposed methods, offering a robust solution for enhancing crab detection accuracy in greenhouse farming systems.

ACKNOWLEDGEMENT

Firstly, I wish to thank Allah for giving me the opportunity to embark on my Master and for completing this long and challenging journey successfully. I would like to thank my supervisors, Dr. Lee Beng Yong, Professor Ts. Dr. Khong Heng Yen, and Dr. Yong Yueh Tiam for their continuous support, patience, and guidance. Their insightful feedback and encouragement have been invaluable. I am also grateful to the College of Computing, Mathematics and Informatics at UiTM Sarawak for providing the resources and environment conducive to research and learning. A special thanks goes to the Sarawak Digital Economy Corporation Berhad, for their financial support through the SDEC Grant. This grant provided me with the necessary funding to carry out my research and explore new dimensions of my topic. I would also like to extend my gratitude to my colleagues and friends who have offered their support, shared their knowledge, and made this journey memorable. Finally, I would like to thank my parents for their unwavering support and encouragement throughout my academic endeavours. Their belief in me has been a source of motivation and strength. Thank you all for your contributions and support. Alhamdulillah.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABBREVIATION

CNN	Convolutional Neural Network
DIoU-NMS	Distance Intersection Over Union Non-Maximum Suppression
k-NN	K-Nearest Neighbour
LDA	Linear Discriminant Analysis
mAP	Mean Average Precision
NMS	Non-Maximum Suppression
PCA	Principal Component Analysis
R-CNN	Region With Convolutional Neural Network
ResNet	Residual Network
RFC	Random Forest Classifier
RNN	Recurrent Neural Network
SVM	Support Vector Machine
t-SNE	T-Distributed Stochastic Neighbour Embedding
YOLO	You Only Look Once Algorithm

CHAPTER 1
INTRODUCTION

1.1 Introduction

The global market for soft-shell crabs is increasing over the years due to their high nutritional value, ease of consumption and unique taste (Waiho et al., 2021). To meet this demand, the industry is increasingly adopting greenhouse aquaculture system. These controlled environments offer protection against diseases. Thus, allow for the regulation both temperature and sunlight. Moulting of the exoskeleton, is a critical process in the growth of crabs (Waiho et al., 2021). The crab shells remain soft for few hours after moulting then it harden again after absorbing water (Fazhan et al., 2022). Since, the new shell hardens in few hours after moulting, crabs must be harvested immediately to maintain their commercial value.

Despite the advantages of greenhouse farming, the harvesting remains inefficient due to a heavy reliance on manual labour. In conventional crab farming, farmers need to check every 2 to 3 hours interval by sliding the crab container which attach with the bamboo at the lake to cultivate moulted crabs (Pitakphongmetha et al., 2021). This method is not only time-consuming but also difficult to scale as farm size increases. to increase productivity, researcher has try to use computer vision to detect crab moulting by using computer vision (Hu et al., 2021), but the experiment is limited to control environment and not proven to be successful in open greenhouse environment.

However, implementing automated detection in an open greenhouse environment presents unique challenges that distinguish it from laboratory or controlled settings (Ghadge et al., 2021).. While previous studies have achieved high accuracy in controlled environments, real-world greenhouse conditions introduce significant "noise" that hinders standard detection algorithms. Specifically, the use of rail-mounted cameras to cover large area with wide-angle lenses, which introduce distortion. This causes the crabs at the edges of the image to appear stretched. Furthermore, the greenhouse structure and water surface create strong glare and reflections from the sun and artificial lighting, which obscures the crabs and complicate detections.

Therefore, the purpose of this study is to develop a pre-processing pipeline to correct wide-angle lens distortion and reduce environmental glare in uncontrolled greenhouse setting. By integrating these enhancements technique, Fisheye Image Correction and 2D-Image Flat-Field Correction with YOLOv7 object detection model. This study aims to provide a robust solution that minimise false detection and effectively automates the labour-intensive process of soft-shell crab farming. The finding of the study is integrated in the computer vision algorithm to detect the moulting event as shown in the Figure 1.1.

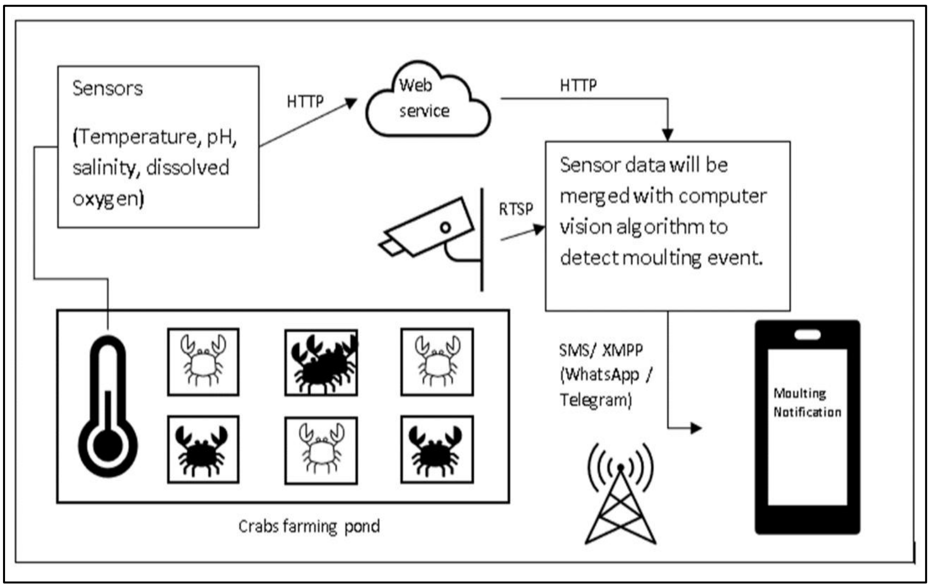


Figure 1.1 IoT Monitoring System for Soft-Shell Crab Farming

1.2 Motivation for This Work

The motivation for developing an IoT monitoring system for soft shell crabs in a greenhouse is driven by the need to enhance operational efficiency, accuracy, and profitability in aquaculture. Traditional manual monitoring is labour-intensive and prone to errors, often leading to missed optimal harvest windows and increased costs. By automating the process with a camera cart, image detection, and real-time notifications via an internet messaging application, farm operators can ensure timely and precise identification of moulted crabs, significantly reduce labour demands and improve yield quality. This integration of advanced technologies not only streamlines the workflow but also maximizes productivity and profitability, making the aquaculture process more efficient and sustainable.

1.3 Problem Statement

Object detection is a crucial task in computer vision, as it allows for identifying and locating objects within an image or video. Detection of crabs, shrimp, and fish under noisy, turbid, or highly reflective underwater conditions is primarily achieved through advanced computer vision techniques, specifically deep learning-based object detection (YOLO series) combined with image enhancement and preprocessing to mitigate environmental interference (J. Li et al., 2023). However, the noise can significantly affect the performance of object detection algorithms thus the image recognition. Environmental factors, such as lighting and temperature, contribute to distortion in the image (Kim et al., 2020). Light overexposure to images can cause loss of detail and introduce noise, making it difficult to accurately classify and recognize objects within the image (Abebe et al., 2018). Specifically, intense glare and reflections on the water surface cause pixel saturation, where high-intensity brightness washes out the unique texture and colour features of the crab. This optical noise creates 'blind spots' or false positives, preventing the feature extraction layers of the detection model from distinguishing the crab from the background. Therefore, exposure and environment are critical factors when working with noisy images. This is because an image's noise can obscure the features of the objects being detected (Zuo et al., 2018). Hence, this study aims to reduce image noise by lowering interference while preserving the original characteristics of the image.

Additionally, object detection can be challenging when dealing with objects that have high similarities (Jiahang Liu et al., 2022). Distinguishing between them can be difficult. As computer vision technology advances, these challenges will be addressed by developing improved object detection algorithms and techniques.

In greenhouse crab farming, capturing images of crabs is essential for monitoring their health and behaviour. Cameras are installed on a cart with rails within the greenhouse, allowing for comprehensive coverage and the ability to move along preset paths. This IoT setup enables wide-angle shots that can capture many crabs simultaneously, providing a broad overview of the population. The cart moves using a microcontroller connected to a network that send the images straight to local computer. However, the wide-angle perspective can introduce challenges such as image distortion which significantly alters the geometric integrity of the objects. This radial distortion causes crabs located at the periphery of the image to appear stretched or curved, changing their aspect ratio and shape. Consequently, the detection model fails to recognise these distorted shapes as crabs because it no longer matches the morphological features learned during training.

These weaknesses of these previous modulations of detection highlighted the need of pre-processing as an essential measure of enhancing the precision of detection of crabs. Images denoising procedures, glare control, and wide-angle review to correct images had to be pre-processed to take care of the noise and other distortions that had to be inherent in the aquaculture environments. Large cameras commonly employed to detect extensive lengths in crab engineering have encountered grave distortions in images at the periphery, and these distortions have the ability to overlap with correct detection (Yang et al., 2020). Also, the presence of glare that water surfaces produced, when using natural light on greenhouse conditions, was another issue to the detection process (Rojek et al., 2023). These issues also embedded the use of specific pre-processing such as, wide-angle image repair and reflected rectifying to improve the quality of the input images prior to applying detecting algorithms.

The initial techniques addressed detection only and lacked complexities in dealing with noisy and distorted images importance of pre-processing techniques has been emphasized. Modern approaches like YOLO offer significant improvements in real-time detection, but their success is still dependent on high-quality, pre-processed input data. This underscores the essential role that pre-processing plays in ensuring the effectiveness of detection algorithms, even in the era of deep learning.

1.4 Research Questions

The primary research question to be answered in this study is:

i. What combination of image pre-processing techniques effectively improves the performance of soft-shell crab detection in greenhouse aquaculture environments?

ii. How effectively does the proposed YOLOv7 model reduce false detections and improve the performance compared to unprocessed images?

1.5 Research Objectives

The study aims to develop and evaluate the IoT-based image detection system for monitoring soft-shell crabs in greenhouse aquaculture. To achieve this, the following specific objectives pursued:

i. To identify the suitable image pre-processing methods for crab detection in greenhouse aquaculture environments.
ii. To evaluate the detection performance of the proposed model under different environmental conditions.

1.6 Significant of Study

This study on image detection, aim to contribute the development of more robust and accurate algorithms not only for soft-shell crab detection. By exploring ways to improve image detection in the presence of noise, this proposal can provide valuable insights into the design of more effective monitoring systems for soft-shell crab farming. Such as real-time detection in a complex and cluttered environment. The significance of a study on object detection for soft-shell crab detection is essential for proper monitoring, management and reduces the mortality rate, but it can be challenging due to the presence of noise such as variations in lighting, distortion and other environmental factors. Furthermore, the study can also contribute to the broader field of computer vision by applying multiple techniques to handle noise and improve image detection in general.

1.7 Limitation

The limitations of this study include several environmental and technical factors that can adversely affect the accuracy of the image detection model. First, inadequate lighting conditions can result in poor image quality, making it difficult for the model to detect the crabs accurately. Second, camera instability or shakiness can lead to blurred images, further complicating the detection process and reducing the reliability of the results. Lastly, the colour similarity between the crabs and their container, especially when the container's colour is murky or faded, can cause the model to struggle with distinguishing the crabs from their background.

1.8 Assumption

This study operates under several key assumptions to ensure consistency and reliability in the image recognition process. Firstly, it is assumed that each container will house only one crab, eliminating the complexities associated with detecting and distinguishing multiple crabs within a single frame. The system is programmed to interpret the detection of two distinct objects within a single container as a positive moulting event. Secondly, all containers used in the study are of identical size and colour, providing a uniform background that helps minimize variability and potential confusion for the model. Next, water inside the container will always be regulated to keep it clean. Lastly, it is assumed that all cameras employed in the experiment are identical and configured with uniform settings, ensuring that any variations in image quality or perspective are due to the crabs and not discrepancies in the equipment or its setup.

1.9 Ethical Committee

The Faculty/State Research Committee has reviewed the study protocol and recommends for exemption from ethical review. UiTM Research Ethics Committee has issued The Ethics Review Exemption with Reference Number: 600-TNCPI (5/1/6) on 2nd October 2023.

1.10 Thesis Scope

This thesis dedicates to better understanding on how image processing and enhancement can improve the object detection accuracy for soft-shell crab. Especially when encounter such issues as glare and reflection that can distort the images.

1.11 Scope of Study

This study aims to improve the object detection for soft-shell crab using YOLOv7 object detection within open greenhouse environment. Specifically centred on improving detection accuracy by addressing the challenges such as wide-angle lens distortion and environmental reflections (glare) caused by sunlight and water surfaces. The model will be trained with a custom dataset that simulates real-world conditions collected from NeoCrab Farm in Lundu, Sarawak. The hardware is defined by a custom

IoT Monitoring system featuring a rail mounted camera cart equipped with standard outdoor CCTV cameras and controlled by an Arduino microcontroller. The crabs are placed in a fixed colour containers to simplify object detection process.

1.12 Thesis Outline

This study explores the integration of object detection for greenhouse crab farming to enhance sustainability and efficiency. Chapter 1 Introduction highlights the need for innovative monitoring solutions to address the challenges of traditional crab farming. Next, chapter 2 literature review examines existing methods and identifies gaps in image-based monitoring, setting the stage for the methodology. In chapter 3 methodology details the development and implementation of an automated monitoring system, including camera installation, data collection, and the training of a YOLO v7 object detection for crab detection. For chapter 4 Results and Discussion, results show the effectiveness in accurately detecting crabs, with a performance evaluation using precision, recall, and F1 score. The discussion compares these findings with traditional methods, highlighting the advantages of the applying optimal technique. Finally, chapter 5 conclusion, the summarises the study's contributions to sustainable aquaculture, acknowledging limitations and suggesting future research directions to further improve image detection.

CHAPTER 2
LITERATURE REVIEW

2.1 Introduction

In recent years, image detection has seen significant advancements, with deep learning algorithms becoming the go-to approach for detecting and localising objects in images and videos (Murthy et al., 2020). This report includes previous studies related to state-of-the-art object detection. After reviewing the earlier studies, the aim is to discuss the challenges, limitations, and future directions for this study. This chapter provides a comprehensive overview and highlights the most recent and impactful contributions to this field.

A study conducted by Baharuddin et al. (2022) proposed a machine-learning classifier method for detecting soft-shell crabs in images. They make a comparison between K-Nearest Neighbour (K-NN), Support Vector Machine (SVM) and Random Forest Classifier (RFC) to study which is the better algorithm for soft-shell crab moulting classification. The study mentioned that K-NN has an accuracy of 98.3%, 96.7% for SVM and 96.7% for RFC. Another study by Hu et al. (2021) proposed a real-time method for detecting soft-shell crabs in images using YOLOv4. The algorithms have achieved a precision of 98.64% in identifying the moulted crab. However, this detection model is limited to controlled environments.

Image recognition is the ability of a computer system to recognize and identify objects or patterns within a digital image or video (Patil, 2021). In the context of the study, image recognition is used to identify crabs in images. This means that the computer system can analyse the image and distinguish whether a crab is present within it. The goal of image recognition is to teach computers to understand and interpret visual information in the same way that humans do, enabling them to classify and categorise images accurately and efficiently (Patil, 2021).

Object detection is the process of identifying the presence and location of objects in an image (Jiahang Liu et al., 2022). It goes beyond image recognition by not only recognising what objects are present in the image but also detecting where they are located. For example, in the context of the study mentioned, object detection involves identifying not only that there is a crab in the image but also where exactly it is in the frame (Hu et al., 2021). This additional information on object location is useful for many applications, such as tracking objects in a video or automating the sorting of images based on their contents.

Prior to the emergence of technologies like YOLO object detection, observing marine life through various instruments fulfilled diverse functions in multiple fields of research and management (Mascarenhas & Keck, 2018). Underwater cameras, hydrophones, and sonar systems, among other devices, were crucial in assessing biodiversity, managing fisheries, conducting ecological research, and monitoring the environment. These tools allowed scientists to thoroughly examine marine environments, ranging from coral reefs to deep-sea habitats, offering vital information on ecosystem vitality, species locations and interactions (Shen et al., 2024). Furthermore, monitoring devices played a crucial role in marine conservation, initiatives, assisting in the tracking of at-risk species, evaluating marine protected zones, and detecting environmental risks (Wang & Wang, 2021). Additionally, these supported oceanographic studies by offering information on ocean currents, nutrient cycles, and the effects of climate change. In aquaculture practices, monitoring tools guarantee ideal conditions for fish cultivation by tracking fish well-being, feeding habits, and water quality.

Greenhouse farming for aquaculture has been utilised in various settings to enhance productivity and sustainability (Ben-Lhachemi et al., 2024; Maraveas, 2023). One of the notable benefits of greenhouse crab farming is a controlled environment where sunlight, diseases and temperature control with the integration of advance technology, allows for precise monitoring of crab health (Nimitkul et al., 2022). By employing sensors and automated systems, farmers can continuously track water quality, temperature, and other critical parameters, ensuring optimal living conditions for the crabs (Pitakphongmetha et al., 2021). Additionally, feeding systems can be automated and adjusted based on real-time data, leading to more efficient and effective feeding practices (Fazhan et al., 2022). A specialised application of this technology is in soft-shell crab farming, where monitoring the moulting process is crucial.

Figure 2.1 represents a structured Soft-shell Crab Detection framework with three major functional pillars: Data Collection, Pre-processing, and Detection. The first step is to address the basic layer of data collection, emphasizing the need to collect images from both underwater and farm environments while overcoming the challenges associated with the dataset. The second major pillar is dedicated to signal refinement, where the use of noise reduction and reflection correction is made to overcome the issues associated with environmental interference. In addition, the use of special geometric corrections like Fisheye Flattening and Flat-Field Correction is made to overcome the issues associated with the camera lens. The final step is the detection phase, where the trade-off between high-speed Single-stage and high-precision Two-stage models is made to ensure the effective detection of the crabs.

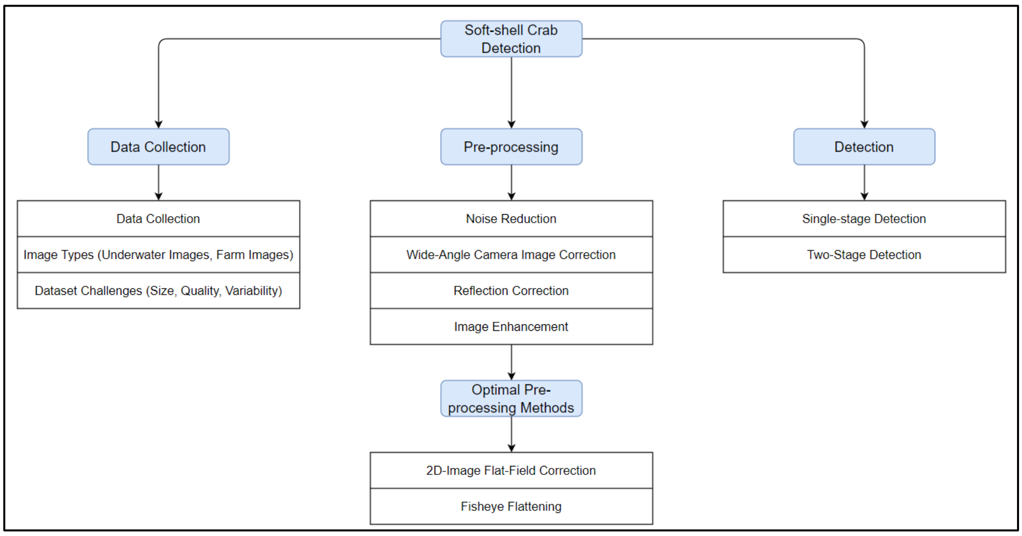


Figure 2.1 Tree diagram for soft-shell crab detection

2.2 Previous Research on Pre-processing Method

The process of image pre-processing is a crucial step in the image detection as it prepares the acquired image for further processing (Shahriar & Li, 2020). Image pre-processing several techniques that are applied to the image to make it more suitable for analysis. The main techniques used in image pre-processing are image denoising, image resizing, and image transformation as listed in Table 2.1

Table 2.1
Common Pre-processing Techniques Used in Object Detection

Technique	**Description**	**References**
Image Enhancement	These techniques aim to improve the visual quality of an image by mitigating unwanted artifacts such as glare and reflections, which can obscure important details and reduce the accuracy of subsequent image analysis tasks.	(Hassan et al., 2024)
Image Correction	It involves adjusting the image to a consistent state or removing distortions, which can help in preparing the image for further processing or analysis. This can include processes like correcting perspective, adjusting lighting, or aligning	(Sneha & Kaul, 2022)

	different layers in the image to ensure uniformity.	
Image Denoising	Removing any noise or irrelevant information from the image to make it clearer and easier to analyse. Techniques used for image denoising include median filtering, Gaussian smoothing, and morphological operations.	(Villar et al., 2017) (Srisha & Khan, 2013) (Deng & Cahill, 1993)
Image Resizing	This involves changing the size of the image to match the size of the training images used for training the machine learning model. This is important because the model has been trained on images of a certain size and resizing the image to a different size may affect its performance.	(Lin et al., 2014)
Image Transformation	Image transforms into a different representation that is more suitable for analysis. Common transformations used in image recognition include converting the image to grayscale, histogram equalisation, and normalisation.	(Hou et al., 2018)

Image pre-processing aims to enhance the essential image features that are significant for analysis and processing tasks, while also suppressing any unwanted distortions present in the image (Shahriar & Li, 2020). Note that these methods tend to be used in a certain order to produce the intended result. The chosen method may differ based on the type of image and the requirement for an image detection.

These environmental noises can be reduced using certain pre-processing techniques by previous researchers. In aquatic environments, images are prone to low visibility, colour casts, and blurriness due to the absorption and scattering of light by water particles. This leads to low contrast images with considerable colour distortion. Colour correction algorithms to normalize the images (Jie Liu et al., 2022). In addition, motion blur caused by the movement of water and floating debris or bubbles often covers the object of interest (Xu et al., 2023). To handle small artifacts Morphological Opening Operations were used to remove noise while preserving the crab’s shape (Srisha & Khan, 2013). In the greenhouse environment, low light conditions often produce Gaussian or salt and pepper noises. These noises can be filter using Mean Filters and using Gaussian/Median filters (Hu et al., 2023). However, traditional techniques such as thresholding cannot be used in greenhouse environments due to uneven illumination caused by shadows.

2.2.1 Type of Noise

Different kinds of noise usually influence the quality of images, including Gaussian noise, salt and pepper noise and Poisson noise. All these are common forms of noise that may break image detection algorithms (Verma & Ali, 2013). The key features are noise reduction methods including filtering and denoising to improve the utility of the image when considering multiple image applications (Pawan et al., 2010).). It is important to know noise types, to use appropriate noise reduction technique and to enhance the quality of image.

Figure 2.2 Original Image (Verma & Ali, 2013)

Gaussian noise is a kind of a random noise distributed normally (Gaussians). When a low-quality camera is used or in low-light conditions in pictures. A Gaussian noise introduces probability distributions on a pixel-by-pixel basis on an image and is capable of blurring and loss of detail (Pawan et al., 2010). Denoising methods such as bilateral or Gaussian filtering may be applied to reduce the impact of the Gaussian noise on image detection.

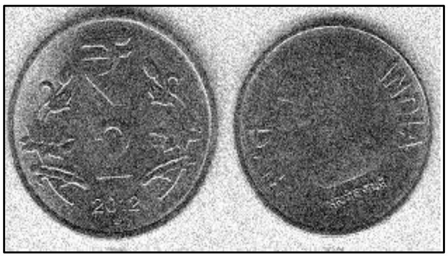


Figure 2.3 Gaussian Noise (Verma & Ali, 2013).

Another form of random noise is salt and pepper noise that may be brought about by such factors as transmission errors or problems in storage of data. It is in the form of randomly dispersed pixel white and black where they appear randomly around the image (Boncelet, 2009). Detecting the noise can be especially difficult to the image detecting algorithms because the noise can lead to either false detection or flawed extraction of features (Mythili et al., 2011). The salt and pepper noise can be entailed by techniques such as median filtering or morphological filtering.

Figure 2.4 Salt and Pepper Noise (Verma & Ali, 2013)

The noise capable of causing, when number of photons sensed by the sensor is insufficient to give detectable statistics, typically does so in medical imaging or astronomical images, is called Poisson or shot photon noise (Pawan et al., 2010). It is caused by random fluctuation of the photon count observed in individual pixels (Agrawal & Raskar, 2009). Poisson noise may also render pictures jagged that cannot be easily detected by the algorithm. Methods that reduce the effects of Poisson noise include wavelet denoising or Poisson maximum likelihood estimation.

Figure 2.5 Poisson Noise (Verma & Ali, 2013)

These are typical image noises, Gaussian noise, Salt and Pepper noise and Poisson noise, which can adversely affect picture detention algorithms. Nevertheless, several denoising methods are at our disposal which can be implemented to alleviate the effects these kinds of noise have on image detection performance.

2.2.2 Image Denoising Techniques

Image improvement techniques such as contrast adjustment, colour correction, image enhancements such as noise reduction and image sharpening among others are essential. The methods serve to deal with the problem of low visibility, colour distortion, and blurriness of light absorption, and scattering in water to improve the input of deep learning models. The papers by (Han et al., 2020; Hu et al., 2021; Yao et al., 2019) exemplified the concept of optimising the underwater image through applying image preprocessing and enhancement methods that allowed detecting the objects more efficiently.

Image noise is a challenging and very crucial issue, which researchers in the computer vision field have drawn a lot of attention. There are a number of sources of noise such as sensor noise, exposure and environmental noise (Verma & Ali, 2013). They may compromise image detection capability to effectively recognize an object or feature in a picture (Nayan et al., 2020).

There is a big effect of sensor noise on the image detection algorithms accuracy. In an investigative article by (Nazaré et al., 2018) a convolutional neural network (CNN) was employed in classifying images of handwritten digits and sensor noise was imposed on these images in different degrees. The model trained on images with salt and pepper noise (p=0.3) achieved the highest accuracy of 93.13% and the best overall result 97.27±1.74%, based on the MNIST multibit data (Nazaré et al., 2018). The results from this study showed that even low levels of noise can reduce the accuracy of the CNN significantly.

Furthermore, environmental noise that is caused by variations in lighting conditions, reflection and occlusion can interfere with image detection quality. To classify imageries under other lighting conditions, the accuracy of the detection algorithm decreased in low-light settings when deep learning-based object detection algorithm was applied (Liu et al., 2021).

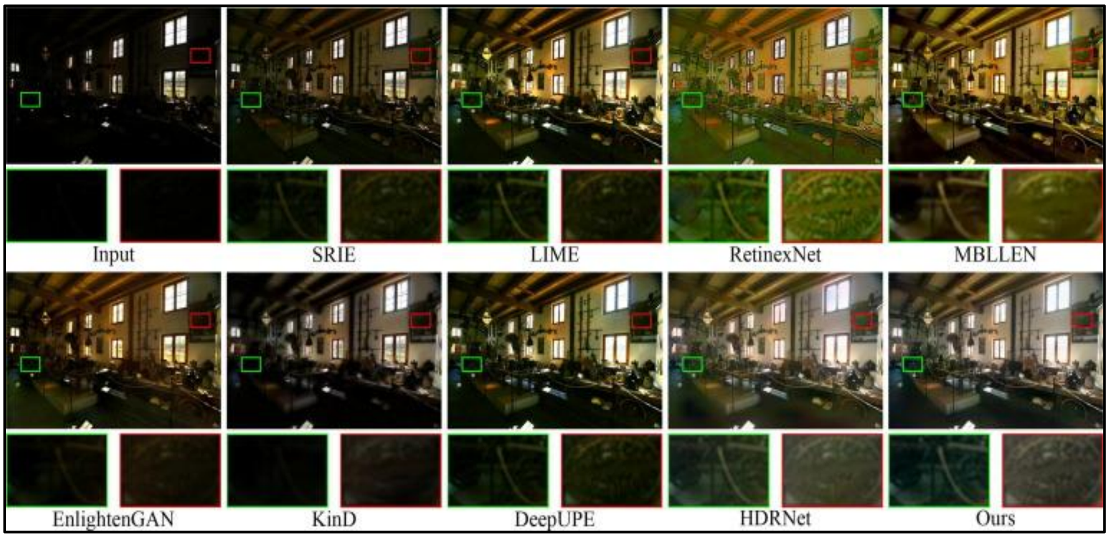


Figure 2.6 Visual comparison on a test image from MEF dataset (Liu et al., 2021).

Image denoising goals to get an accurate clean signal of the specific underlying signal. Different applications that require it include image compression, image restoration, image segmentation, and object detection (Fan et al., 2019).

Image denoising has been done with many methods. For example, linear filtering methods such as mean filtering, median filtering, and Gaussian filtering. In mean filtering, the values of each pixel will be substituted with the average value of the neighbouring pixels (Shakoor & Tajeripour, 2015). Median filtering involves the replacement of the value of individual pixels with a median density of images (Villar et al., 2017). Gaussian filtering involves the replacement of the value of each pixel with an average of adjacent pixels based upon a Gaussian distribution (Deng & Cahill, 1993).

Another method is the non-linear filtering that incorporates median-mean filtering, adaptive median filtering and bilateral filtering (Yaroslavsky, 2004). A median-mean filtering is a combination of the median and the mean filtering wherein the median filtered image is computed to calculate the output in the mean filtered and vice versa (Banerjee et al., 2015). The adaptive median filtering is the addition to median filtering that aims at adapting to the image structure (Soni et al., 2019). Bilateral filtering is a non-linear filtering method whereby, a dimensional distance given a Gaussian function is utilized to weight the pixels along with the difference of intensity of the pixels (Kornprobst et al., 2009).

Image denoising methods based on deep learning used convolutional neural networks (CNN) and recurrent neural networks (RNN). A type of neural network used to process grid like structures like images is called CNN and a type of neural network used to process sequences of data like time series data is called RNN. Image denoising has been implemented both CNN and RNN with CNN being more frequently used (Zuo et al., 2018).

The morphological opening operator is used to remove tiny size objects within an image, in this case, the bubbles in an image by putting in place a structuring element into the image (Srisha & Khan, 2013). The element size and shape can be modified to attack an intended type of objects in the image. It is possible to apply it with the help of a structuring element, which determines the shape and size of the neighbourhood of a pixel not overlooked during the operation (Srisha & Khan, 2013). The filtering effect can then be tuned to fit the size and shape of the noise in the image to the extent that one can vary the size and shape of the structuring element.

Image denoising is a crucial process when it comes to eliminating undesired noise in an image, yet the subject should be maintained at the same time. Image denoising can take on a variety of methods, such as the traditional methods and deep learning methods, with the latter types being more common since they can effectively produce a deep learning of the underlying structure of the image and tele-purr the presence of noise.

2.2.2.1 Wide-Angle Camera Image Correction

The image capturing in green house aquaculture farms usually uses wide-angle lenses. This allows total capturing of the spacious environment, with the focus on the wide operation. Furthermore, the clutter of containers and equipment, which is hard to cope with the use of standard lenses, is brilliantly managed by the wide-angle lenses, using which one can see all details on the image for the system process it better. Even the wide-angle lens invokes an impression of expansiveness, pre-empting the potential physical confinement of the equipment. Distortion is the main issue in which straight lines on the edges of the image are made to bend and curve. It is quite possible to consider a busy background as tangled and devoid of a subject matter. In addition, foreground objects in close-ups may seem too enormous or deformed unless they are correctly situated. The distortion might be less evident in the images of aquaculture in greenhouse. However, close compositions should be achieved to avoid cluttered background and foreground features.

Research and development have been a prime focus of investigation in all sorts of applications, and it has exceeded the boundaries of marine life imagery (Hu et al., 2021). The rectification of wide-angle lenses introduced distortions in the images is critical to generating accurate and visually faithful representations of the scenes in various fields including computer vision and photography (Fan et al., 2019; Yang et al., 2020). Researchers have used different correction techniques to address the overall problem that is a result of distortion in wide angle cameras. The lens distortion correction is a main technique employed in image correction of the wide-angle camera (Alemán-Flores et al., 2013). The wide-angle lenses often introduce distortions, such as those of barrel-distortion, in which straight lines are observed to deform away and / pincushion distortion/width distortion, where lines are observed to deform toward each other (Abebe et al., 2018). Such distortions can greatly affect the visual clarity of images, especially when the camera angle strays from the central axis. Earlier research developed techniques to correct these distortions (Yang et al., 2020) without losing the accuracy of the object shapes, sizes, and spatial relationships in the environments (Jiahang Liu et al., 2022). In photography and cinematography, wide-angle distortion removal enhances the aesthetic beauty of video or photo, thus resulting to a more desirable and life-like depiction of scenes (Shih et al., 2019). Moreover, correction methods are crucial in surveillance, robotics and augmented reality, where it is required to clear up the correct understanding and localization of objects in the scene.

Camera angle effects on images are also a significant issue when it comes to wide-angle camera (Chang et al., 2022). During the realignment of lenses in relation to the centre axis, the deformation is more pronounced, which influences the quality of the images in general (Chang et al., 2022; Hu et al., 2021). These effects must be recognised and considered, especially in fields where to measure them precisely or to represent them exactly what is difficult. Some of the most common solutions have been proposed in attempting to fix distortion created by very wide lenses in photographs. One of the prevalent algorithms involves camera calibration to measure intrinsic camera parameters such as focal length, distortion coefficients (Shih et al., 2019). Techniques that employ the pattern of calibration and the mathematical models such as the pinhole camera model, are well investigated. The open CV library provides the Calibration camera resources and distortion correction (Fischer et al., 2021). By calibration of cameras, and utilizing of distortion algorithms, researchers have achieved great achievements in the correction of wide-angle lens distortion thereby enhancing the accuracy of pursuit computer vision processes.

2.2.2.2 Rectifying Reflection

One of the most common issues when it comes to taking pictures in greenhouse aquaculture farming is the problem of reflections due to diverse surroundings as well as the materials and constructions employed. Greenhouse aquaculture refers to a farming method where the aquatic animals, such as fish, shrimp, or crab, are kept or reared in controlled conditions, typically by enclosures of glass or plastic. These materials, intended to allow sunlight to enter while preserving warmth, exhibit strong reflectivity, leading to light reflecting off surfaces and generating glare or reflections in photos taken.

Moreover, reflection has been worsened by the presence of water in aquaculture systems. Undisturbed and calm water reflects objects situated near it or a building. Furthermore, the use of artificially lighting in greenhouses intensifies the issue of reflections since the use of various sources of light such as, overhead lamps or green house lights, may result in glare or hot spots, when reflected on water surfaces, or the walls of the greenhouse.

- Adjusting Camera Position: Changing the camera's position away from the reflective surface can reduce the direct reflections. Adjusting the camera's angle also can decrease the issue with light source, reflective surface, and camera lens.
- Applying Polarising Filters: Polarising filter can reduce the glare and reflection by filtering specific light wavelengths. Mounting polarising filter on the camera lens helps to reduce reflection and improve image sharpness.
- Controlled Lighting: Controlling the position and the temperature of the light in the greenhouse can help reduce the reflection. With proper arrangement, this can minimise the reflection by reducing the exposure of the light directly into the container.
- Diffusing Light: Installing diffusers or soft boxes helps to disperse the light and reduce the sharp reflections with more consistent lighting conditions. Soft lighting can reduce the reflection in the container. The image taken will look more natural and clearer.

Recent reports have focused on creation of new process of compensating reflection and glare in images. Deep-learning methods have also shown promise of autonomously discovering and removing reflection and glare artifacts. Trained CNN-driven networks on labelled data demonstrated the capability to derive specific complex patterns in the context of reflections and effectively reduce them (Zhang et al., 2021). Moreover, researchers examined the use of polarized filters and methods of multi-image fusion to mitigate the effects of reflection and glare (Yao et al., 2019). It is possible to capture multiple images relative to the various polarization levels or exposure levels, and the separation of reflected objects and the underlying scene needs to be carried so that one can reconstruct the images without glare. Pre-processing tools that remove reflections and enhance the indentation of images can be particularly beneficial to algorithms of object detection, recognition, and tracking (Sun et al., 2023). Adding reflection and glare correction modules to existing computer vision pipelines have helped researchers to improve autonomous driving, surveillance, and robotics performance.

Flat-field correction is a crucial administrative phase in digital imaging that addresses gaps in illumination and sensor which helps ensure a high quality of images. Studies mentioned the importance of providing flat-field images under controlled light conditions to generate accurate master flat-fields, by which to correct (Kokka et al., 2019). Furthermore, another research showed the success of flat-field correction in minimizing noise and enhancing contrast in medical imaging contexts (Sayed et al., 2023). Furthermore, the role of flat-field correction in enhancing image quality in astronomical research, which is an indication of its applicability in revealing subtle facts about the star structures (Kokka et al., 2019; Zheng et al., 2023). In the remote senses, it is shown that flat-field correction is effective in the accurate classification of land cover and observation of the environment (Fischer et al., 2021). With the combination of these research works, it is evident that flat-field correction is critical to amplifying image quality and accuracy in a variety of disciplines suited to pose valuable insights into its implementation and benefits.

The non-local means (NLM) filter transformed the domain of image denoising (Heo et al., 2020). This method results in enhanced denoising results, especially in maintaining edges and textures while successfully eliminating noise. The NLM filter's effectiveness is more than general denoising. It has revealed itself to be useful in various image processing tasks, such as deinterlacing, whereby the NLM filter can be applied to fill in missing lines in interlaced video frames, the result of which brings about a better quality of video (Ali, 2018). Next, there is the interpolation of views. The approach uses the NLM filter to generate middle viewpoints between captured images in furtherance of creation of smooth transitions in the encoding of multiple view videos (Dehghannasiri & Shirani, 2013). The NLM filter can be used to regularize depth maps to refine them by removing noise and enhancing quality (Ibrahim et al., 2020). The NLM filter has been a useful approach in eliminating noise in pictures in addition to other image processing applications. Its ability to remove noise effectively retaining image details makes it an effective tool in addressing various image process tasks.

Dimensionality reduction is a critical procedure in the detection process as it attempts to reduce the number of levels of the data content but leaves behind the significant data that is pertinent in the recognition (van der Maaten et al., 2007). The purpose of doing this is to ensure efficiency of the recognition process as well as to accelerate the speed of the recognition. Dimensionality reduction aims at reducing high-dimensional feature list into a low dimension state with the same combination of the relevant information (van der Maaten et al., 2007). The dimensionality reduction technique adopted is determined by the issue under consideration, size of the data set, and output to be provided. Visualizing high-dimensional data in a two-dimensional or three-dimensional space is typically applied to PCA (Mishra et al., 2017), whereas image compression and data visualization are typically applied to LDA (Shashoa et al., 2016). t-SNE is often used for visualizing high-dimensional data in a two- or three-dimensional space (van der Maaten & Hinton, 2008).

The image detection can lower the computational requirement and memory of the data, hence decreasing the memory and cost of the computation, with dimensionality reduction, which enables the detection process to be more efficient (Rakhshanfar & Amer, 2018). This action is significant to the whole system's performance because the quality of the output depends on the quality of the input.

2.2.2.3 Optimal Methods for Pre-processing

Two pre-processing methods used in crab detection, 2D-image flat-field correction and Fisheyes Image Correction, have been useful in alleviating the problems associated with wide-angle lens distortion and glare/reflection. Such techniques greatly improve the level of images acquired in the greenhouse aquaculture settings, where the conditions of variable light are prevalent along with large field-of-view systems.

The first technique, 2D-image flat-field correction is developed to offset the non-uniformity in light coverage of the image, commonly produced by the natural light settings in open greenhouse aquaculture. The overexposure in some and under exposure in others can make it hard to tell crabs in such environments according to the detection algorithm (Maraveas & Bartzanas, 2021).). Flat-field correction is based on balancing the intensity variations throughout the image, in effect equalizing the lighting to allow all areas of the image to be consistently exposed. It is especially effective in eliminating glare on water surfaces that outperform brightly in the centre and under expose the edges, as well. Flat-field correction will also make the picture more even as far as brightness is concerned, which enhances the effectiveness of detection algorithms (Wen et al., 2024).

The second technique, Fisheyes Image Correction, addresses the distortions issue produced by wide-angle or fisheye lenses, which are often used in capturing areas of aquaculture environments. Although these types of lenses offer an excellent field of view, they create high degrees of geometric distortion, especially at the fringes of the image with objects appearing stretched or distorted (Yang et al., 2020). It is possible that this distortion can strongly affect the results of detection since crabs at the edges might not be appropriately recognized because of their distorted morphology. These distortions are corrected with the help of Fisheyes Image Correction methods, i.e., lens calibration and distortion mapping, which transform the image mathematically to ensure that the proportions of objects are the correct ones (Gai et al., 2024). Flattening the image ensures that crabs appear in their true form regardless of their position in the frame, allowing for more precise detection across the entire image (Iftikhar et al., 2024).

Together, 2D-image flat-field correction and Fisheyes Image Correction form an optimal pre-processing pipeline that enhances image quality before applying detection algorithms. By addressing issues of uneven illumination and geometric distortion, these methods provide a clearer, more consistent dataset for machine learning models or traditional image processing techniques. Improved image quality is crucial for ensuring the accuracy and efficiency of soft-shell crab detection, particularly in environments where lighting conditions and lens distortions would otherwise compromise the detection process (Chen et al., 2021).

2.3 Previous Research on Detection Methods

Some methods introduced in the literature have been used in the past to detect soft-shell crabs, starting both with the classic image processing algorithms and further up to the more advanced machine learning algorithms. Early methods had almost entirely involved manual and primitive steps toward image processing, including thresholding (Prathusha et al., 2018) and edge detection (Almarinez & Hernandez, 2019). The limitation of the methods prompted by the sensitivity to given environmental conditions, such as lighting differences and noise, was easy to implement. An example of this case was the use of thresholding, which failed to perform well in situations where the light reflection or shadows covered the crabs and, thus, gave an erroneous result in the segmentation of the image. Equally, edge object-detection techniques were not always able to discriminate between the crabs and the environment, particularly within aquaculture environments where the water surface interaction, as well as univocity of lighting produced multifaceted image conditions. Though applicable on doing simple jobs these methods were not strong enough to be used in the real-time and have automated detection of crabs in a changing environment.

Technology emerged, and the more advanced machine learning techniques, including Support Vector Machines (SVM) and K-Nearest Neighbours (KNN) were introduced into object detection (Baharuddin et al., 2022). SVM was promising since it could classify crabs through learning by using labelled data, and it could be considered more accurate than the conventional methods. SVM however was disadvantaged by the fact that it utilized massive feature extraction which necessitated human intervention to determine the feature of the image which was of interest such as texture and shape. This reliance on feature engineering constrained the scalability of SVM when used on large sizes of data or under conditions associated with features as in the aquaculture farms. Likewise, although KNN was relatively simple in its implementation, it was computationally inefficient and costly with larger data sets to operate in real-time. Both SVM and KNN, as much as they were better than previous methods, yet they posed some difficulties in a noisy environment when images were subject to reflections or distortion caused by the wide-angle lens (Yang et al., 2020).

Deep learning specifically with the Convolutional neural networks (CNNs) and with one-stage detection YOLO (You Only Look Once) innovated deep learning-based methods of detecting the presence of deep well shell crabs (Hu et al., 2021). These techniques did not require feature extraction done manually, instead the deep learning models could automatically extract features necessary including learning the appropriate features to deep learning data images. The object-detection feature of YOLO, which is possible in real-time, proved particularly useful where it was necessary to detect crabs quickly and accurately. Nonetheless, even with the strength of deep learning techniques, the quality of the input data is also decisive. Poorly pre-processed images caused by problems such as noise, reflections, or distortion of a lens may adversely affect the performance of the best detection algorithms even. Pre-processing is therefore a very crucial process within the detection pipeline, as it makes sure that the data that is fed to these models is free of flaws at least good enough to support accurate and realistic identification.

The concept of the gradual shift of traditional image processing algorithms to machine learning and deep machine algorithms supports the claim that the need to identify the presence of soft-shell crabs remains problematic to this day (Tang et al., 2020). Image detection has seen tremendous progress over the past few decades, thanks to advancements in computer vision and deep learning. Table 2.2 provides brief overview of the major milestones in the development of object detection:

Table 2.2
The Development Of Milestones For Object Detection

Development Milestone	Description	Year	References
Traditional Methods	Object detection based on handcrafted features and traditional computer vision algorithms	1999	(Lowe, 1999)
Viola-Jones Algorithm	Object detection using Haar-like features and AdaBoost, achieved real-time face detection	2001	(Viola & Jones, 2001)
HOG and SVM	Object detection using Histograms of Oriented Gradients features and Support Vector Machines, achieved state-of-the-art results	2005	(Dalal & Triggs, 2005)
Deep Learning	The introduction of deep neural networks, particularly Convolutional Neural Networks, revolutionised object detection	2012	(Krizhevsky et al., 2012)
Faster R-CNN	Object detection framework using a region proposal network and a CNN, achieved state-of-the-art performance	2015	(Ren et al., 2015)
One-Stage Detectors	Object detectors such as YOLO and SSD that are fast and simple, widely adopted in applications such as autonomous driving and robotics	2016	(Redmon et al., 2016)

2.4 Crab Detection Method

2.4.1 Single-Stage vs. Two-Stage Detection

Object detection takes two major directions single-stage and two stage detection. They each have their own benefits, but they are vastly different in their complexity, speed and accuracy and the selection between one or the other is important regardless of the functions of the application.

Single-stage detection, such as You Only Look Once (YOLO) algorithms, is object detection that uses a single neural network, shown to run in one pass by combining object localization with object classification (Redmon et al., 2016). The primary advantage of YOLO is its speed; by skipping the proposal stage used in two-stage methods, YOLO can process images in real-time, making it highly suitable for applications where quick detection is essential, such as monitoring crabs in aquaculture environments (Bochkovskiy et al., 2020). YOLO splits the input image into a grid and concurrently produces predictions of bounding boxes and class probabilities of every grid cell, allowing to quickly perform end-to-end detection.

In contrast, two-stage detection, including R-CNN (Region-based Convolutional Neural Networks), and its derivatives (Fast R-CNN, Faster R-CNN) follow two different stages. The former step creates proposals of the regions, determining possible locations of objects, and the second step is the classification of objects in those regions (Girshick, 2015). Two-stage architectures are only known to be more precise in challenging scenes and on small objects but are significantly slower than single-stage bottlenecks such as YOLO thus less suited to a real-time system (Ren et al., 2017).

In real-time soft-shell crab detection, it is mainly the efficiency requirement that drove the choice to use YOLO versus two-stage methods. Fishing conditions, including soft-shell crab farms, often require real-time monitoring, to record important events, such as the moulting period. The single-stage detection architecture of YOLO is exclusively adopted to this task, being much faster than two-stage models and at the same time offering a competitive accuracy (Wang et al., 2022). This speed is vital for processing large volumes of images in real time, where delays could mean missing critical events.

Moreover, the architecture of YOLO balances speed and accuracy offering solid detection performance even in tasks that are more challenging e.g. underwater or greenhouse aquaculture where the lighting conditions and distortions complicate detection. The ability of YOLO to find crabs in the image, instead of concentrating on collision areas, is directly because it looks at the image entirety which guarantee that crabs within the image are located in detail (Hu et al., 2021).

Although two-stage models such as R-CNN provide greater accuracy when using small objects or in a complex environment, their slow processing speed makes them less feasible in real-time. Where speed and precision are not important, such as in soft-shell crab farming, the compromise between speed and accuracy is very much in favour of YOLO which offers the best performance and efficiency ratio. The need to focus on the single-stage YOLO in detecting soft-shell crabs is justified by its high potential to achieve real-time and adequately high accuracy, which is why it is a perfect choice to implement there.

2.4.2 Detection

2.4.2.1 Single-Stage Detection YOLO (You Only Look Once)

YOLO (You Only Look Once) is a real-time object detector that has undergone several release versions, i.e. YOLOv1, YOLOv2, YOLOv3, YOLOv4, YOLOv5, YOLOv6, and the most recent is YOLOv7, as of this study. Each version have brought both intrinsic gains in the level of detection accuracy, speed, and efficiency (Bochkovskiy et al., 2020; Redmon et al., 2016; C. Y. Wang et al., 2022).

Unlike a two-stage detection system, such as R-CNN, which initializes region proposals before then classifying them, YOLO is a single stage detector. Localization and classification are both done in one forward pass over the neural network giving YOLO greater speed and efficiency on a real-time streaming task where running time is critical (Redmon et al., 2016). YOLO splits the input picture into a grid directly estimating bounding boxes and the probabilities of classes per grid cell. The design specializes in processing images in real-time, thereby making it suitable to suit environments such as crab detection in aquaculture (Hu et al., 2021).

The consideration of a single-stage detector, such as YOLO, is especially useful in situations related to real-time detection due to its nature of higher speed and reduced computational costs. Real-time processing is important in soft-shell crab detection, in which cameras easily record images to identify moulting crabs, and respond promptly, by deploying actions or alerts. Consequently, two-stage detectors, which may be more accurate, add a delay to the system because region proposals are computed followed by classification, implying that they are not applicable to an instant decision-making scenario (Baharuddin et al., 2022).

YOLO algorithm operates based on breaking an image into a grid of grid cells and associated with every cell, the algorithm predicts the bounding boxes and class probabilities of the objects the cell may have. In contrast to other object detection-based methods, such as object detection by sliding a window over the image, or using a region proposal network, the object detection process in YOLO operates all the pixels of an image in one forward propagation of a convolutional neural network. That is why the application of YOLO is extremely fast and effective, and real-time object detection is feasible on the devices with limited resources (C. Y. Wang et al., 2022).

Moreover, YOLO has numerous applications making it one of the significant touches of the specific field. For starter, computer vision. YOLO has had a significant application in computer vision activities like object detection, image segmentation and image detection. It has been employed in identifying objects in an image and video, including cars, people, animals, objects in surveillance cameras and self-driving cars (Redmon et al., 2016).

Furthermore, YOLO also can be applied in agriculture and aquaculture. YOLO has been used for various tasks such as detecting and counting fruits, vegetables, and weeds in images and videos captured by drones or cameras. A study Häni et al. (2018) used YOLO to detect and count apples in orchards. The same goes for aquaculture, YOLO has been used to detect and classify fish and many aquatic lifeforms in images and videos captured by cameras or drones. Muksit et al. (2022) implemented the method of study based on YOLO to identify and classify fish in fish farms. The other study by (Hu et al., 2021) applied YOLOv4 to identify crab moulting. The works revealed that YOLO may help crabs or crab counts with precision and on-the-fly.

Overall, YOLO has shown promising results in real-time object detection and classification. However, it's important to note that YOLO is just one of the many algorithms that could be used, and its performance will depend on the specific application and dataset. Figure 2.7 show the network architecture and Table 2.3 show the evolution of YOLO Object Detection.

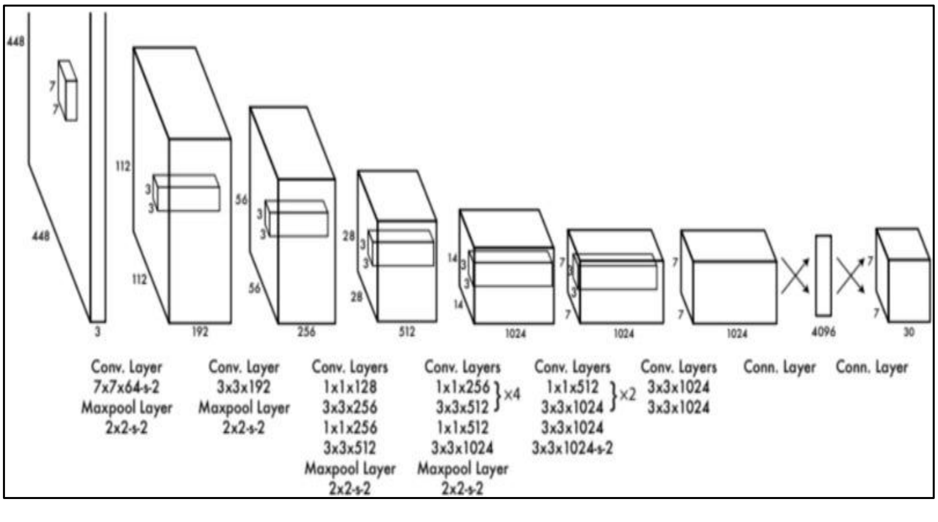


Figure 2.7 YOLO network architecture (Murat & Kiran, 2025)

Table 2.3
The Evolution of YOLO Object Detection

YOLO Version	Year	Features	References
YOLOv1	2016	Based on a single convolutional neural network (CNN) architecture that performed object detection in a single pass, making it faster than other object detection methods at the time. However, it had limitations in terms of accuracy and the ability to detect small objects.	(Redmon et al., 2016)
YOLOv2	2016	Introducing anchor boxes to improve detection accuracy, as well as a high-resolution classifier to better handle small objects. This version also introduced the concept of prediction across multiple scales, thus improving the detection of objects of different sizes.	(Redmon & Farhadi, 2016)
YOLOv3	2018	Further improvements by using a deeper network with more layers and a new anchor box design. It also introduced a technique called "feature pyramid networks" (FPN) that improved the detection of objects at different scales. YOLOv3 achieved state-of-the-art accuracy at the time, but the detection speed was still slower than some other real-time object detection methods.	(Redmon & Farhadi, 2018)
YOLOv4	2020	Using a more efficient architecture, a new training technique, and a new set of anchor boxes. It also introduces new detection heads that can detect objects in a more robust and accurate way, as well as using a technique called "Cross-stage partial connections" (CSP) that allows the network to maintain a high level of accuracy at high frame rates.	(Bochkovskiy et al., 2020)
YOLOv5	2020	A new anchor-free approach for object detection which eliminates the need for anchor boxes, resulting in faster and more accurate object detection. Data augmentation technique called Mosaic data augmentation which improves the robustness of the model to object occlusion and scale variations. Improved training pipeline that allows for faster training and inference times. YOLOv5 is the first version of YOLO that can be run on edge devices, due to its small memory footprint and high inference speed	(Jie Liu et al., 2022)
YOLOv6	2022	SPP-YOLO (Spatial Pyramid Pooling YOLO) which improves the accuracy of object detection by using a spatial pyramid pooling layer to aggregate features from multiple scales. Improving training pipeline that utilizes a new data augmentation technique called CutMix, which helps to improve the model's robustness to object occlusion and scale variations. The use of a new loss function	(Li et al., 2022)

		called Focal loss which helps to improve the model's performance on class imbalance datasets.	
YOLOv7	2022	SPADE-YOLO (Spatially Adaptive Dimension-wise Spatial Feature Ensemble YOLO) which improves the accuracy of object detection by using a dimension-wise ensemble of features from multiple scales. It allows the model to better adapt to different objects, scales, and environments. Using an improved training pipeline that utilizes a new data augmentation technique called MixUp, which helps to improve the model's detection to object occlusion and variation of scale changes. This allows the model to better generalize and perform well on unseen data.	(C. Y. Wang et al., 2022)

2.4.2.2 Two-Stage Detection

The common machine learning algorithm that is utilized in the detection and classification of images is Support Vector Machines (SVM). SVM is a linear classifier, which is then used to divide the data into various classes by identifying the optimal boundary otherwise referred to as the decision boundary which divides the data into various classes. In doing this, the highest possible distance between the classes is maximized and the joint result is the boundary between a decision zone and the nearest data points by each class (Cervantes et al., 2020).

SVM operates by scaling up the data to a higher dimensional space by a feature mapping task at which the information can be drastically segregated by a linear boundary. After locating the boundary, it can be applied to those (data) that are unknown to the classes and assign them to the predefined classes. SVM can be applied when the data could not be separated linearly in its initial form since the feature mapping capable makes it possible to separate the data in the upper dimensional space (Wang et al., 2021).

In the context of image detection, SVM would enable the categorization of images into various images, including animal, vehicle, or scene (Gaye et al., 2021).). The initial step in applying SVM to image detect is feature extraction of the image in terms of edges, textures and shapes of the image. The features are then entered into the SVM algorithm to provide the SVM with inputs and train a model to locate the decision boundary (Cervantes et al., 2020).

One advantage of SVM is that it can work with a big volume of data, not only a big, but with high-dimensional data as well, thus suitable to detect images. Noise and outliers in the data do not affect SVM either because the optimal algorithm only uses the nearest data points to construct the decision boundary (Gaye et al., 2021). This makes it more reliable and accurate than other machine learning algorithms that can be distorted by data noise, such as K-Nearest Neighbours, that may be affected by noise in the data (Nugrahaeni & Mutijarsa, 2016).

However, SVM can be sensitive to the selection of the kernel function employed in the feature mapping process, as well as the regularization parameter which is employed to control complexity of a model. Existence of a kernel function that is not adequately matched to the data can give the decision boundary that is suboptimal, and an inappropriate level of the regularization parameter may cause overfitting and underfitting, respectively (Nedaie & Najafi, 2018).

Despite these challenges, SVM is one of the powerful means of image recognition and it has been extensively applied in many applications, among them facial recognition, detection of objects and classification of scenes (Le et al., 2012). Over the last few years, the deep learning models; specifically Convolutional Neural Networks (CNN), have contributed to large improvement of image detectors, outperforming SVM in a variety of tasks (Wang et al., 2021). SVM is still one of the widely and successfully used methods of image detection especially when there are limited computational resources.

The SVM transforms the values into a temporarily faster dimension (capacity to sample a multidimensional plane) so that a linear structure is identified between and among the data values and its groups. The SVM can filter noise and data outliers, which is why it is a stable and highly precise instrument in detecting images. But this may be sensitive to the selection of the kernel functional along with regularization factor and the performance of SVM can be overtaken by the deep learning algorithms in certain applications.

Convolutional Neural Networks (CNN) are a type of deep learning algorithm that has been widely used in various fields and applications (Alzubaidi et al., 2021). CNN is particularly well-suited for image and video processing tasks, such as image classification, object detection, and semantic segmentation.

Image classification is one of the most popular uses of CNN and consists in attaching a label to an image according to its contents (Alzubaidi et al., 2021). CNN has been demonstrated to perform state-of-the-art on some image classification datasets, up to the ImageNet (Krizhevsky et al., 2012).

Some other common uses of CNN also include object detection which is detecting and locating objects in an image or a video. Similar methods of object detection have been developed on CNN, including R-CNN (Girshick et al., 2014), Fast R-CNN (Girshick, 2015), YOLO (Redmon et al., 2016), and SSD (Liu et al., 2016). Such techniques have been used in self-driving cars, video surveillance and robotics.

Semantic segmentation has also been conducted on CNN and entails labelling each pixel of an image. This enables a finer-grained interpretation of the contents of an image and has been used in activity-tasks like autonomous driving (Chaitra et al., 2020).

CNN have also been applied in other applications like natural language processing and speech recognition in addition to processing images and videos. As an example, machine translation has been enhanced through CNN on CNN (Kalchbrenner & Blunsom, 2013) as well as speech recognition (Abdel-Hamid et al., 2014).

In general, CNN have turned out to be quite effective and generalisable deep-learning algorithms that can be applicable in a variety of applications. Therefore, have demonstrated state of the art performance in image and video processing applications and have been used in other domains including self-driving cars, video surveillance, robotics, natural language processing and speech recognition.

One of the simplest type of algorithm in the field applied to detect images is the k-Nearest Neighbour (k-NN) algorithm, which is an example of instance-based learning aspects that do not necessitate explicit training of the model (Guo et al., 2004). Instead, it makes use of the raw data to do some detection activities. The simplest concept of k-NN is to identify the k closest neighbours of a particular test case in forensic space and to assess the cluster labels of these neighbours to forecast the cluster label attribute of test case. K is the cost of the nearest neighbours the prediction relies on and a significant algorithm parameter (Taunk et al., 2019).

According to the study carried out by Fan et al. (2021) the k-NN algorithm is normally used in the feature space derived as a result of application in an image data. Depending on the image processing methods, the features can be determined with regards to edge detection, texture analysis or shape analysis. As soon as the features have been extracted, it will serve as a foundation of the k-NN algorithm. Upon being given a new test image, k-NN algorithm calculates the distance between features of the test image and that of each training image. The nearest neighbour of the images of the test picture is determined as the k training pictures having minimal dissimilarities to the image, and the forecast of the category of the image in the category of the closest neighbours is completed with the majority of the ballots of the category of the closest neighbours (Fan et al., 2021).

The simplicity of k-NN is also the key benefit. It has low costs to calculate and minimal parameter tuning (Nugrahaeni & Mutijarsa, 2016). In addition, it is very flexible to new data, and it manages noisy data well. This ensures that it is an attractive option to most image detectors, particularly when the sample sizes are extensive or the data considered is vastly diverse.

Nevertheless, there are also limitations related to the k -NN algorithm. This fact has its principal demerits namely its high memory requirement. The size of that whole training set is required to be stored in memory and, therefore, k-NN cannot be used with large image detection requirements when the size of the training set cannot be stored in memory. As well, k-NN is sensitive to k value and the effectiveness of the algorithm can run with varying k choices (Wang et al., 2021).

A basic but useful algorithm to detect an image is k-Nearest Neighbour (k-NN). It is efficient with low-level problems and using noisy or diverse data. Its high memory needs and susceptibility to the selection of k, however, render it inapplicable to large-scale image detection problems.

2.4.2.3 Training Data for Machine Learning Image Classification

Machine learning image classification algorithm needs to be given a training dataset to which the algorithm owns to learn and ascertain patterns in the image information. Then, the test dataset is used to test the best model of the training. It is with this that the algorithm may be used to categorize new images under the corresponding categorizations.

Data splitting is an essential step in machine learning which in this case focus on image classification. It involves dividing a dataset into two or more subsets for training, validation, and testing purposes (Joseph & Vakayil, 2021). The optimal proportion between meaningful data the model is given in machine learning can be different based on the size of the data and the complexity of the subject matter being encountered and also based on the particular issue (Muraina, 2022). Nonetheless, the most typical way is to divide the statistic into training and testing samples of 70:30 or 80:20. In general, the training set should contain the majority of the data (e.g., 70% or 80%), while the testing set should contain the remaining data (e.g., 30% or 20%) (Joseph & Vakayil, 2021). This makes sure that the model has enough data to learn on while training and that one has a large enough testing set to test the performance of the model.

2.5 Applications of Image Detection

The image detection, which is a fundamental area of machine learning and computer vision has transformed many industries by raising the level of efficiency and accuracy in areas where it is involved. It is applied in the automotive sector to the autonomous cars and the system, responsible of advancing driver-assistance, thus making safer navigation (Kanagaraj et al., 2021). In agriculture, image detection helps precision farming and livestock promotion, tracking the condition of resource-treated and the overall condition of livestock to identify which resources aid the recovery (Tian et al., 2020). Facial recognition and intrusion detection systems are used to improve the level of security and access control (Owayjan et al., 2013). In manufacturing, it is used in quality control and automation, and this guarantees high quality and productivity (Maraveas, 2023). Wildlife conservation and pollution detection initiatives are among the benefits of environmental monitoring (Machado & Chen, 2019). It is used in the smart cities in serve to manage traffic and safety of people to enhance the infrastructure of the city and emergency management (Maraveas & Bartzanas, 2021). Also, it supports interactive learning and enhanced scientific work in education and research (Xu et al., 2023). Overall, image detection is a flexible technology that is propelling innovation and attaining solution to complicated issues in a range of industries.

2.5.1 Underwater Object Detection Using Deep Learning

The ability to detect objects underwater can be identified as one of the primordial research domains, driven by its use in marine biology, environmental science, and underwater robotics due to its adaptability, its speed, and its precision (Xu et al., 2023). Indicatively, Wang et al. (2021) combines the variants covered with CSPDarknet53 network and Spatial Pyramid Pooling (SPP) to boost detection accuracy and speed, achieving better outcomes on objects in water with mAP of 41.2%. Similarly, as is in the work by Muksit et al. (2022), the use of YOLO models to identify fish species under realistic underwater conditions proves to be multi-functional and achieves a substantial level of accuracy and effectiveness. Other approaches, like integrating attention controls and blending multi-scale characteristics, have received examination to enhance ability of detection in harsh underwater conditions. Embedding a criss-cross system global interaction-focused selective attention mechanism into YOLO models has been demonstrated to increase meaningful features leading to a high-detection accuracy of objects (Shen et al., 2024). The refinements of the YOLO algorithm in underwater environments highlight its potential as an important tool in the study of marine biodiversity, marine conservation, and marine exploration providing an efficient solution to accurate real-time recognition and monitoring of marine life.

Images taken underwater are normally of bad quality due to the absorption and attenuation of the light at the water as well as the reflected light at the surface. This affects the sharpness and accuracy of the image that, in turn, affects the work of the object recognition algorithms. Xu et al. (2023) studied the challenges of underwater imaging conditions which include a low contrast and substantial colour distortion, which affects the accuracy of detection models. Jie Liu et al. (2022) formulated an approach of improving the underwater images by applying a series of options in improving the expression of reflection and lighting parameters of the image by colour correction.

Motion blur in underwater images may be caused by deflection of water, the camera or the subject (Mascarenhas & Keck, 2018). Moreover, the object of interest can be covered with floating particles (Chen et al., 2021). Such factors might complicate image detection to get accurate detection.

Deep learning models require large volumes of data to successfully train. The scarcity of such datasets to underwater processes hobbles the capability of the models to learn intricate characteristics and procedures. This will compromise the model accuracy and precision. Although annotated underwater data exists, there is likely an issue of class imbalance whereby one or two species or objects will be over-represented with the rest not adequately represented. The outcome of this imbalance may be poor model performance. Mumuni (2022) noted that data augmentation is especially important because it can be used to add variety and scale the amount of training data.

For applications such as monitoring and counting autonomously underwater, detection algorithms need to operate in real-time. The computational demands of deep learning models (such as YOLO) as noted by Bochkovskiy et al. (2020), can be the limiting factor and also emphasises the importance to optimise processing speed while maintaining accuracy.

Limited number of training samples for network training can result in overfitting (Shorten & Khoshgoftaar, 2019) and to overcome the lack of annotated underwater datasets, researchers have utilised dataset augmentation and synthetic data generation. *Cutmix* have been used to enhance images (Yun et al., 2019). To increase the variety of training data for underwater detection models, methods such as image augmentation with multiple configurations (rotation, flip, scaling) were used. C. Y. Wang et al. (2022) enhance the types and number of target categories across various water environment with mosaic data augmentation and Mix-up data augmentation method. Maharana et al. (2022) applied Online Dataset Preprocessing (ODP) enhancing the detection process with improved sample number and enhance the model complexity.

Optimizing YOLOv4 for both speed and accuracy was important to meet real-time processing requirements for underwater object detection. Bochkovskiy et al. (2020) made the advancement thus enhancing the computational efficiency of deep quantization and development of lightweight model architecture.

P. Li et al. (2023) suggested a dedicated modification of YOLOv4-Tiny network that with it balanced speed performance and smaller framework integrated with and Efficient Channel Attention (ECA) module made it applicable to deployment in real-time aquaculture areas. recommend that underwater objects, which have been detected and identified following real-time detection, be offered as B-YOLOX-S and used in the fishing operations. It possesses small parameters and can be deployed on underwater vehicle easily to carry out the task. Zhang et al. (2021) suggested MobileNet v2 as a backbone has excellent accuracy to speed trade-off.

YOLO is a single-stage object detection framework that operates solely in one network to perform not only the recognition of objects but also their classification at the same time. It is a single step method and typically yield faster and light than most other detection (C. Y. Wang et al., 2022). Over time, the original YOLO has undergone successive improvement, with each addition extending what could be relied on in matters of speed and precision. As an illustration, the YOLOv2 added features, including improved performance such as batch normalization, resolution (as in Faster-RCNN) and reference boxes (detectors/localisers) (Redmon & Farhadi, 2017). The bounding box prediction was also optimized and feature extraction network enhanced in YOLOv3 and as such, it was less predictive and more precise than older versions

(Redmon et al., 2016). Then came YOLOv4, which prioritized additional training based on more intelligent training approaches referred to as Bag of Freebies that consisted of data augmentation, rebalancing biased data and optimizing the loss function to regression of bounding boxes (C.-Y. Wang et al., 2022). With YOLOv5, a Cross Stage partial (CSP) bottleneck was introduced to enhance feature extraction (Jie Liu et al., 2022). YOLOv6 extended to subdivision of the backbone and bottleneck layers, which has improved the performance (Horvat et al., 2023). YOLOv7 was concerned with better gradient back propagation, and this enabled the model to learn better (C. Y. Wang et al., 2022). Every YOLO variant implements its architecture with various architectural differences as well as size differences. In general, models with more parameters (and larger) achieve higher detection results but the compromise is that they are even more computation-intensive meaning slower predicted behaviour. One of the considerations that determine which version of YOLO to apply to a particular application is struck on the balance between the model size, its accuracy and efficiency. Such developments demonstrate the ways in which underwater object detectors based on the use of YOLO and other related frameworks are addressing numerous issues. They outline the gradual developments and the flexibility of deep learning methods in underpins marine science and observation.

2.5.2 Image Detection in Soft-shell Crab Farming

Soft-shell crabs are a rapidly developing business, and the moulting is one of its most important phases when crabs cast off their previous shell to let a new one grow. Timing is crucial to farmers because they are interested in harvesting the crabs when the shells are soft (Waiho et al., 2021). Currently, the farmers are inspecting each crab by hand, which is a tedious task that leaves a high chance of missing the optimum harvesting schedule too. To overcome such, researchers have come up with image detecting algorithms that have the capability of automatic identification of moulting. These systems detect minor variation in colour and texture in digital images of the crabs which warn of the occurrence of moulting among crabs (Hu et al., 2021). Using this technology, farmers can scan their crabs in a faster and more specific manner, which will eventually increase efficiency and profitability.

Several studies have been conducted by researchers to come up with and test image detecting algorithms to help work with detecting moulting in soft-shell crabs and the outcome of the research has proved to be highly encouraging. An example is where various machine learning classifiers were used to support the occurrence of moulting in crabs housed in boxes (Baharuddin et al., 2022). The findings showed that the K-Nearest Neighbors (KNN) algorithm outperformed both Support Vector Machine (SVM) and Random Forest Classifier (RFC). KNN achieved an accuracy of 98.3%, while SVM and RFC each reached 96.7% (Baharuddin et al., 2022). Based on these by these findings, it can be recommended that KNN is the most suitable of the classifiers to use in recognition of moulting in soft-shell crab farming.

Another study uses YOLO object detection to detect moulting crab (Hu et al., 2021). In the study, the detection model is based on the YOLO v4 model. The precision of the moulting crab detection reached 98.64%. This study used Key-frame Extraction Method of Inter-frame Difference with Mean Filter because there is no public dataset for soft-shell crab. The study uses video and applying the method to capture the images for training and testing. The actual collected data set cannot be as rich as the public data set, making it challenging to improve the generalized performance of the model and combat overfitting.

To expand the data set, this paper uses geometric and colour transformations, such as image brightness, saturation, contrast, and image noise, randomly. The authors divide the data set into four types: only normal crabs, only moulting crabs, normal crabs and crab shells with high overlap, and normal crabs and crab shells with low overlap. This class ensures that the training samples obtained by the model are sufficient and the test set can fully reflect the performance of the model (Hu et al., 2021).

YOLO v4 model with an enhanced NMS maximum suppression method is used for the detection of moulting crabs, without increasing computational requirements. This improves the model's accuracy and stability, reducing the chances of missed detections while extracting moulting videos (Hu et al., 2021). As shown in Figure 2.8, Hu et al 2021 detect either a crab is moulting or not by detecting either there are two crabs in the image of not. The system relies on good target detection model to improve the non-maximum suppression algorithm based on DIoU-NMS (Distance Intersection over Union - Non-Maximum Suppression).

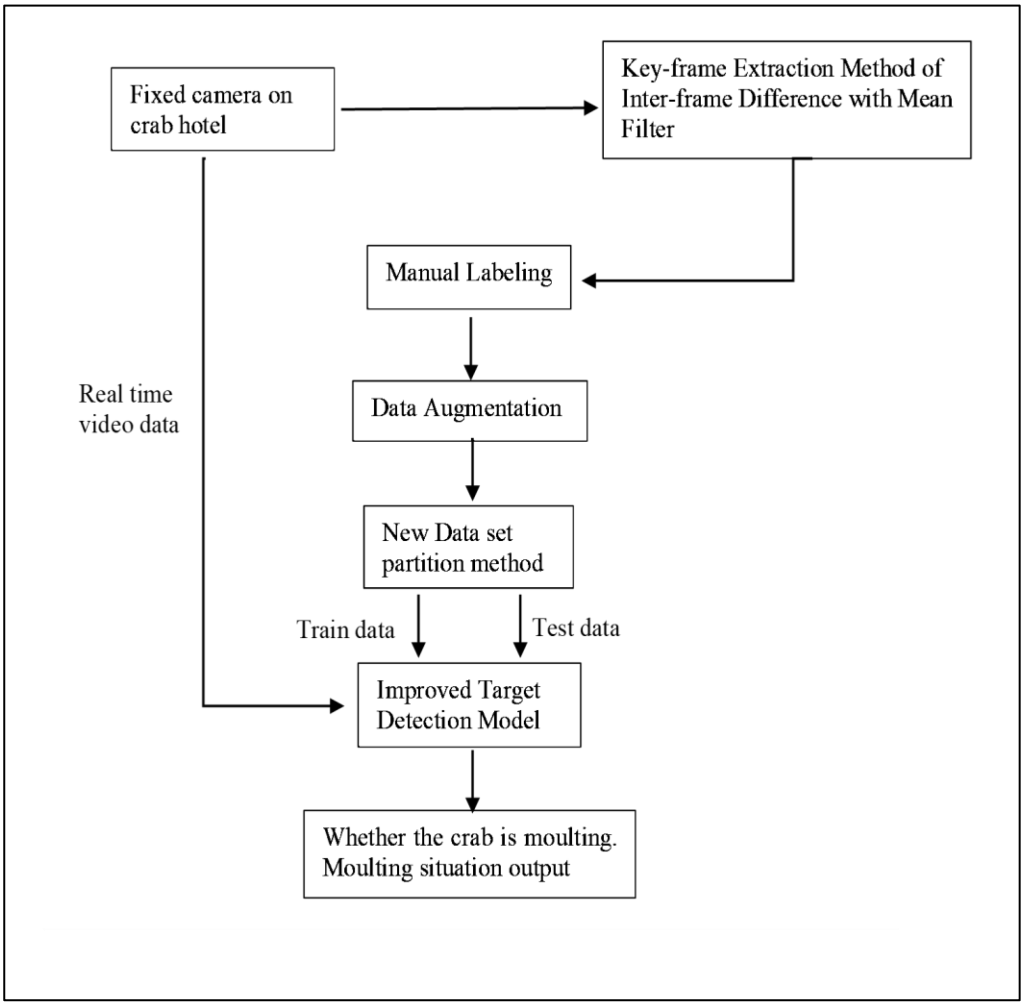


Figure 2.8 Research Framework for detection (Hu et al., 2021)

The experiment was done in a controlled environment to eliminate unnecessary noise to optimise the detection. One of the unsolved issues mentioned in this study is data set in dark and narrow environment of the moulting crab. Image detection algorithms have the potential to revolutionize the way that soft-shell crab farming is done. Future research should focus on testing these techniques on larger samples of crabs and in different environmental conditions, as well as exploring other potential applications of image detection in soft-shell crab farming.

After reviewing previous methodologies, there were clear trade-off between computational cost & accuracy. For example, KNN outperformed SVM (98.3% vs 96.7% accuracy) on small dataset. But it lacks the ability to process complex noisy image in real-time. Deep learning in the other hand, like YOLOv4 achieved higher precision (98.64%) but required more data & augmentation to prevent overfitting. As YOLO series evolved, newer version like YOLOv7 introduced anchor-free approaches and adaptive feature that improved detection accuracy and processing speed compared to older version. Even though YOLOv7 requires more computational power than lightweight models, its ability to balance high-speed inference with accurate detection in noisy dynamic environment makes it most optimal choice for this study.

2.6 Internet of Things

Internet of Things (IoT) is the new powerhouse which has transformed many industries by making it possible to connect many machines and systems together. The idea of the IoT is that physical objects are equipped with sensors and software and other technologies that can gather information and communicate it to support advanced automation and real-time tracking. This review discusses the development of the IoT, its use, issues and opportunities, basing on real-life investigations and academic publications.

Internet of things was the first term adopted by Kevin Ashton in 1999 (Kramp et al., 2013), who imagined a world whereby normal things increase their connection to the internet and therefore easily communicate and exchange data. With the progress of wireless communication, sensor and data analytics over the years, the field of IoT has become a practical technology rather than a theoretical option. According to Alymani (2025), IoT as a group of designed computers with observed and interconnected instruments capable of sensing, communicating and interacting with their surroundings which results in change of the partaking of information in the information processing and utilization.

IoT has been used in many fields, which has contributed greatly to other fields of healthcare, agriculture, smart cities, and manufacturing. Wearable sensors and remote monitoring systems are also a type of IoT devices that have facilitated treatment to patients through constant control of health status and early access to symptoms (Aladdin et al., 2026). IoT has also helped agriculture by using the approach of precision farming that involves making use of sensors and data analysis to improve the crop module and the utilization of resources (Akhter & Sofi, 2022). Smart cities use IoT to help in the effective management of urban facilities such as traffic congestion, waste disposal and power distribution (Alymani, 2025). Manufacturing is a field that embraced IoT to detect maintenance before its failure, increase the efficiency of supply chains, and increase the overall business efficiency (Anozie et al., 2024).

Another area that the IoT has positively impacted is agriculture where it has been used in precision farming to make better use of crops and their resources by employing sensors and data analytics. IoT devices observe soil moisture, temperature, and the amount of nutrients, and these measurements enable the farmers to make fact-based decisions regarding irrigation, fertilization, and pest management (Kumar et al., 2024). This leads to more efficient use of resources, reduced environmental impact, and increased crop productivity.

Likewise, the use of IoT in aquaculture has improved tremendously. In aquaculture, IoT systems are water quality, automated feeding in aqua culture, and health monitoring of aqua species. In real-time, sensors test parameters, including pH, oxygen and temperature and provide ideal living conditions to the aquatic organisms. Automated feeding based on the data can modify the feeding schedule as well as the volume, which boosts the rate and lowers the waste (Son & Jeong, 2024).

Despite its widespread adoption, IoT faces several challenges that need to be addressed to realize its full potential. The main agenda should be security and privacy because more interconnected devices obtained raise the chance of cyber-attacks and data breach. According to Blessing et al. (2024), effective security systems, as well as encryption of data, must be in place to secure sensitive information. The other notable challenge is interoperability where different devices and platforms must effectively cross-communicate with one another. Moreover, data generated by the IoT devices is inexpensively enormous, which creates issues of data storage and handling, and analysis, requiring sophisticated solutions of big data analytics and cloud computing technology (Alymani, 2025).

IoT can be discussed as having a bright future because research and development continues to study the current challenges of the IoT and advance its capabilities. Integration of artificial intelligence (AI) and machine learning (ML) and IoT is likely to make the IoT even smarter and autonomous and predictive analytics and more complex decision-making processes (Ullah et al., 2025). Its implementation on 5G technology will further make the IoT relevant as it will undergo quicker and more stable connectivity that will require real-time applications. Besides, the development of edge computing will allow the processing of data closer to the source to extend their latency and bandwidth consumption (Alymani, 2025).

2.6.1 Hardware setup of IoT

When it comes to the implementation of the IoT in aquaculture, various hardware devices are used to scan and control the aquatic ecosystem successfully. The sensor network is one of the main hardware items that is made of many kinds of sensors that are installed in the water to test the main environment parameters in real-time: temperature, pH, dissolved oxygen, salinity, and ammonia levels. These sensors play important roles in ensuring the proper conditions to support aquatic life and commonly operate to survive in the extreme and unpredictable environmental characteristics of the aquatic environments, such as corrosion and biofouling. High quality and durable sensors are the basis for dependable data collection, which is crucial in the real-time monitoring and control of aquaculture systems.

Another essential supplementary hardware in an IoT-based aquaculture system is the cameras. Under water cameras give a visual follow up of fish behaviour, condition, and feeding behaviour. Such cameras are usually linked to the IoT system through wireless or wired networks and are strategically located at the aquaculture plant. Cameras that have either night image or thermal image capabilities enable round-the-clock footage even in low light condition. The visual information collected through these cameras may be processed through computer vision algorithms to identify disease signals, stressor/abnormal behaviour, and apply early intervention to prevent the possibility of incurring major losses.

The node that controls the rest of the IoT system is usually an Arduino microcontroller or the like, which handles the functioning of the sensors, cameras, and other devices incorporated in the system. The microcontroller takes care of processing the data received by the sensors, its uses in the control of automated functions like feeders and water pumps, and the transfer of information to the central server or cloud computing that will be analysed further. Depending on the data the sensors retrieve, motors and actuators, through connections to the microcontroller operate various mechanical parts, including feeding systems or water circulation devices. Also, it comprises a router or any other networking equipment that is required to create a stable communication channel between the IoT equipment and data processing/data storage centres, providing the transmission of data in real time, as well as the option to be remote and monitor. This built-in hardware will enable to manage efficiently and automatically their aquaculture plants and thus maintain the best conditions of aquatic organisms with the least possible human intervention.

The IoT hardware used in this study consists of two well-placed cameras, an Arduino microcontroller, a railing system, a motor, and a router to provide an internet connection. Images of crabs both vertically and horizontally are taken by the synchronized cameras which have been calibrated to achieve identical settings. The board used as the main controller is the Arduino that controls the cameras and sets the direction of the movement of the containers in the railing to capture an image each time in a very precise and constant manner. When setting the motor within control of the Arduino it moves the containers gently beneath the cameras. The router allows wireless access which gives the transmission of real-time data to a centralized server to store and analyse data. This consolidated system has provided viable, automated and dependable image acquisition and data control of the crab scrutiny software.

2.7 Summary

The previous research on image detection to detect soft-shell crab moulting was performed under controlled environment (Baharuddin et al., 2022; Hu et al., 2021). In these studies, by Baharuddin et al. (2022) and (Hu et al., 2021) the images were taken in perfect conditions, in the absence of noise or intrusion. Uncovering the crabs when they are going through the moulting process may be harmful to their mortality and moulting process. To moult and regrow their shell, crabs need some amount of darkness. Direct light may disrupt their regular circadian cycle, resulting in stress that may raise the probability of mortality and prolong the moulting cycle (Chen et al., 2021). To effectively remove these specific noise types, the most suitable denoising technique is 2D-Image Flat-Field Correction, which mathematically normalizes the light intensity of the image, thus eliminating the "hot spots" of bright intensity that mask the features of the crabs without causing blurring of the edges, which would be the case with other filters. Furthermore, Fisheye Image Correction is used to remove the "spatial noise" or distortion at the edges of the image, ensuring that crabs placed at the edges of the container are not stretched, preserving it shape for the YOLOv7 model to accurately distinguish the features of the crab from the background.

Thus, the environmental conditions and lighting of crabs need to be considered during taking crab photos to infer the existence of a detected object which guarantees the safety and well-being of crabs. Therefore, the algorithms formulated in this research could not be effective in the field where it must work in real world conditions and have more variability. The following chapter will be discussing how this study carried out.

CHAPTER 3
METHODOLOGY

3.1 Research Framework

This study focuses on improving image detection for soft-shell crab detection. The research framework is divided into three processes which are identifying issue and problems, collection of data and finalisation.

Identifying issues and problems includes a background study that highlights the problem statement, which is the noise and high similarity object that can disrupt the object detection. The objectives of this study are to find a suitable method for reducing noise and improving object detection by using pre-process images. The literature review covers the detection algorithm used by previous studies, the limitations of those studies, and the type of noise. The anticipated issue for this study is the detection of crabs at the edge of the container in the image and the glare and the reflection of the crab image.

Data collection involves the methodology used in this study, which includes image acquisition, image augmentation, image pre-processing, classification using YOLOv7, output, and using a dataset that will be collected from the farm for this study.

Finalisation consists of data evaluation, discussion, and conclusion. The data collected will be analysed and evaluated to determine the effectiveness of the image detection. The discussion will highlight the strengths and limitations of the study, and the conclusion will summarize the findings and recommendations for future research. Table 3.1 shows the research framework for this study.

Table 3.1
Research Framework

Process	Activities
Identify Issues and Problems	Background Study: • Soft-Shell Crab IoT Monitoring System Project • Focus: detection of soft-shell crabs moulting process in uncontrolled environment. • Deployment of system at the farm • Aims to develop efficient and accurate method for detecting soft-shell crabs in moulting process. • Suggest appropriate techniques for reducing the reflection and wide-angle lens distortion. Problem Statement: • Reflection and distortion can disrupt the image detection. • Objectives:

		• To assess the influence of image pre-processing techniques to detect the crab in various conditions. • To minimize the false detection between soft-shell crabs and other undesired objects in pre-processed images, thereby improving the accuracy and precision of soft-shell crab detection.
	Literature Review:	• Previous study was using YOLOv4 for detection. Was done in controlled environment to minimise noise. • The limitation for this study is detection in uncontrolled environment. • Object detection model: YOLOv7.
Model Development	Methodology:	• Phase 1: Preparation. • Phase 2: Method • Phase 3: Evaluation
Performance Evaluation		• Data evaluation • Discussion • Conclusion

3.2 Phase 1: Preparation

3.2.1 Problem Analysis

Identify the key challenges in detecting soft-shell crabs. Discuss issues such as distortion from wide-angle lenses, lighting variations, and reflections that affect image quality.

3.2.2 Experiment Setup

To evaluate how different image processing techniques (flattening, reflection/glare reduction) influence the model's detection performance based on the metrics mentioned. Compare the detection performance at the edges and in the middle of the image to understand the impact of wide-angle lens distortion and potential reflections within the containers. Crab on the edge of the image detection 4x1 (4 boxes): Divide the side of the image into four regions for analysis. This helps assess variations in detection accuracy across the area. Crab on the middle of the image detection 3x3 (9 boxes): Divide the middle of the image into nine regions for analysis. This allows for evaluating the detection performance within the container area.

3.2.2.1 *Onsite Testing Greenhouse Setup*

The greenhouse is an open environment where there is light comes through from outside as shown in Figure 3.1. Crab containers are arranged with 4x20 for this study. A wide-angle camera is attached to a custom cart that moves above the crab containers with a rail length of 13ft for image acquisition. The height of the camera is 3ft above the crab container. There is also a shaded canvas to cover direct sunlight above the moving cart (Aini et al., 2025).

This study makes use of the pillar in the greenhouse for the setup. That is why the camera is not centre enough to cover all the boxes. The cart moves every 30 seconds to stop and take the pictures. There are 7 predetermined positions of the container for the cart to stop and taking pictures. After the cart finishes its first trip, it will reverse and start taking pictures from the endpoint to the starting point. Even so, due to the condition of the farm during the experiment, there were not many crabs present in the container. To solve this problem, data augmentation were used to diversify the images.

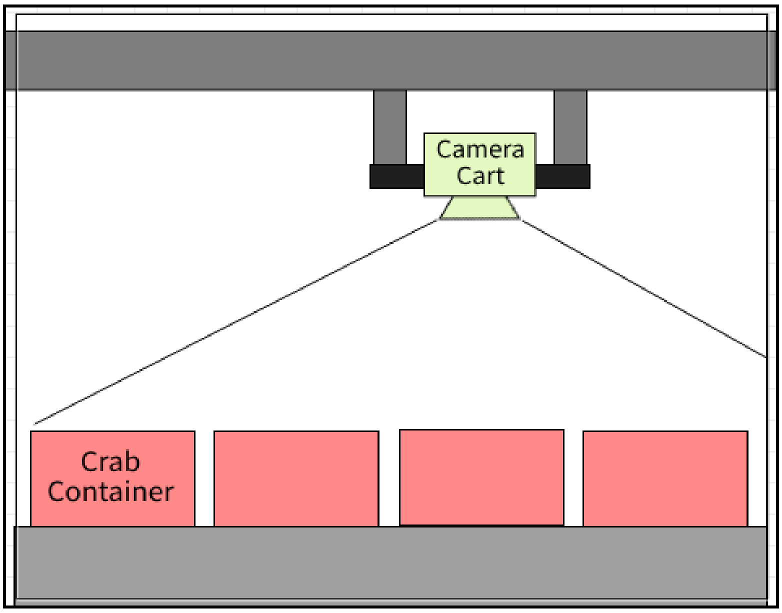


Figure 3.1 Camera Cart Greenhouse Setup

3.2.2.2 *Setup of IoT*

The IoT setup for this study integrates a streamlined hardware configuration to enable real-time monitoring and automated detection of moulted soft-shell crabs in a greenhouse aquaculture environment. The hardware includes two wide-angle lens cameras, an Arduino microcontroller, a motor-driven railing system, and a router for data connectivity. The wide-angle lens cameras are strategically placed to capture a broader field of view, ensuring precise and consistent imaging of the crabs as the containers move across the section. The Arduino microcontroller acts as the central control unit, coordinating the operation of the cameras and the motorized railing system. This ensures the containers are moved with precision along the predefined path, allowing accurate image capture.

The motor-driven railing system is designed to automate the movement of crab containers through different sections, enabling comprehensive monitoring without manual intervention. The router provides wireless connectivity, facilitating the seamless transmission of captured images and data to a computer for analysis. As an added feature, a Telegram Bot is integrated into the setup. This bot is programmed to automatically send images of detected moulted crabs to a designated Telegram group or user. As soon as the system identifies a moulted crab using the YOLO detection model, the image is instantly shared, providing timely updates and alerts to farm operators. But for this experiment, the images were taken directly from the computer at the farm.

The use of wide-angle lens cameras ensures that the system can capture more extensive visual data, minimize blind spots and enhance detection accuracy. This IoT system offers a reliable, automated solution for crab monitoring, reducing manual labor and improving operational efficiency. The integration of automated image capture, real-time data transmission, and immediate notifications ensures effective farm management while providing scalability for larger aquaculture operations.

3.3 Phase 2: Method

This phase shows the flow of the method used in the experiment. Starting with image acquisition to the evaluation. This study also highlighted the image preprocessing where the objectives of this experiment reside.

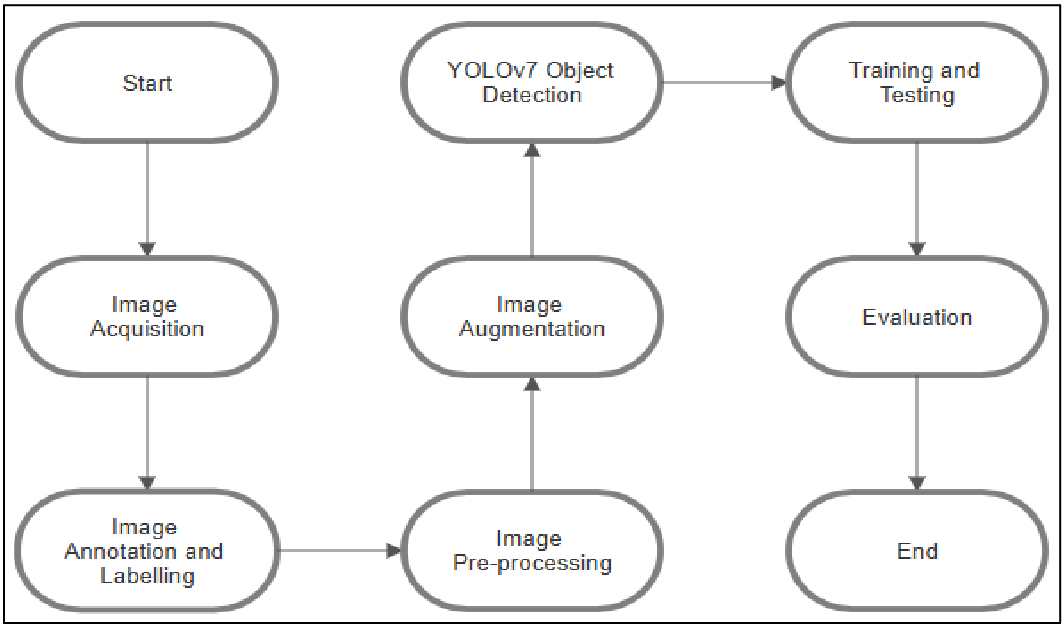


Figure 3.2 Method flowchart

3.3.1 Image Acquisition

The dataset was acquired by IP camera from the moving cart as shown in Figure 3.4. The image data that will be used for training is the crab from the farm. For preliminary testing, a crab replica will be put in the crab box. The position of the crab and everything inside the box will be varied as shown in Figure 3.3. In Figure 3.4, that is how the image was taken with the IP camera at the farm. On the farm, there are only one crab per container.

For the starter, 100 images will be taken and split to 80:20 ratio for training and testing to create a custom dataset. In object detection, 80% of the available data was used for training the algorithm, and 20% used for testing its accuracy. This is done to ensure that the algorithm is trained on enough data to learn complex features accurately and tested on new data to evaluate its accuracy (Tang et al., 2017). The 80:20 ratio can be adjusted based on the dataset's size and complexity and the specific algorithm used. The resolution used for the experiment was 640x640 with sRGB colour format and saved as JPEG. Images taken for the testing will be increased by 100 depending on the optimisation of the algorithm.

Figure 3.3 Lab scale for image detection in preliminary testing

Figure 3.4 Gathering images from the site

In this study, a new primary dataset was used for the object detection of soft-shell crabs moulting. The reason for using a new dataset is due to the unavailability of a public dataset that has been used in previous studies. To create the primary dataset, soft-shell crab will be placed in a container with a pipe that may cause false detection. Images will be taken using IP cameras from multiple angles and under varying lighting conditions to capture the necessary variations in the images.

3.3.2 Onsite Image Acquisition

The image data was taken at the crab farm located at NeoCrab Farm, Lundu, Sarawak, using a wide-angle camera to capture a broader view of the crab containers. The camera specification is a 2.8mm fixed lens, 103° horizontal field of view, and 1920x1080p resolution. For every 30 seconds, the cart will move and then stop to capture the images. The images captured will be focused on showing 4x3 boxes. Images will be transferred using Telegram for convenient data collection. There are 300 images taken but only 150 images were chosen because there is no crab and during the data collection the farmers cleaned the containers. After that, all images taken will be resized to a standard size of 640x640 pixels to ensure consistency for the model. The dataset possesses several challenges that may impact the performance of the detection model. First, the murkiness of the water in the container. Second, the presence of glare during daytime. However, the features of the crab could be distinguished as long it does not hide in the dirt in the container which sometimes it cannot be seen.

Figure 3.5 Sample image taken by the camera

Figure 3.6 The situation where the cart used to carry the cameras is placed on top of the crab containers

3.3.3 Image Annotation and Labelling

For the proposed method, images were annotated and labelled using bounding box annotation for detection algorithms. Annotations were made with the Roboflow tool. Every crab, including those experiencing occlusion, was considered in the annotations. Then, the annotated and labelled data were exported to YOLO format before proceeding for detection.

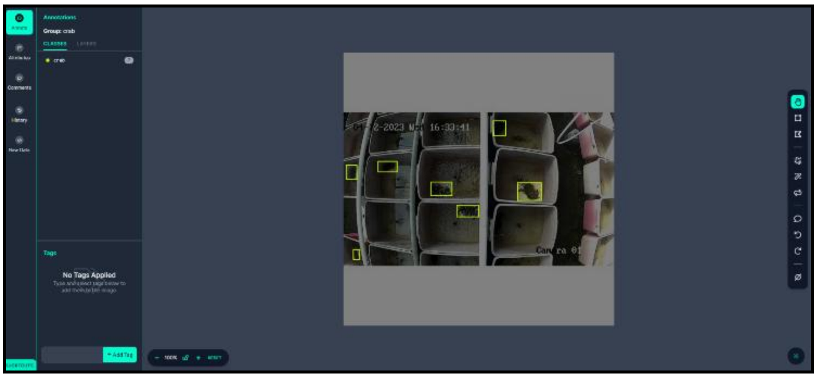


Figure 3.7 Annotating and Labelling Images Using Roboflow

3.3.4 Image Pre-Processing

The images taken will be undergo pre-processing after the images in the dataset resized and labelled them suitable for further processing. The pre-processing technique used in this study include flattening technique (Fisheye Image Correction and stretch) and reflection reduction (2d-image flat field correction and non-local mean filter).

For Fisheye Image Correction, it is a mathematical rectification technique designed to correct the radial distortion caused by the 2.8mm wide-angle lens. The algorithm constructs a specific Camera Matrix (K) with specific parameters, where image dimensions are specified in pixels, focal lengths of fx=640 and fy=600 and setting the optical centre at (cx=320, cy=320). The parameters are then used to correct the image and map the pixel coordinates, effectively "flattening" the shape of crabs in the image periphery, giving a correct shape of the crabs for easier detection.

The "Stretch" method is a geometric correction technique aimed at determining whether complex correction is needed or not. Unlike the Fisheye Image Correction, which corrects the lens curvature using complex mathematics, this algorithm only forces the resizing of the raw input image and compresses it to fit a standard resolution of 640x640 pixels, which is required by the YOLO model. Although this ensures a full frame, it does not correct lens curvature, i.e., objects will be stretched, making this method suitable as a performance benchmark against Fisheye Image Correction.

2D Image Flat-Field Correction is particularly chosen to address issues of glare and lighting inconsistencies arising from the greenhouse setting. Flat-field correction is a method where a "flat-field" image is taken under constant lighting conditions, and then a division operation is performed on each pixel of the obtained raw image and the flat-field image. In this case, an intensity factor of 15 was used to equalize the lighting, effectively cancelling out the high intensity "hot spots" or glare on the water surface, which often hide the position of the crabs.

Non-Local Means (NLM) Filter is a denoising algorithm, and it is used here to compare its effectiveness with Flat-Field Correction in addressing the glare problem. As opposed to other filters, the NLM algorithm attempts to average similar "patches" or sets of pixels within the entire image, rather than individual pixels, to reduce noise and retain edge features. In this case, the parameters of the algorithm were adjusted to represent the patch size, window size, and strength, respectively, to address local glare issues, although it was noted to be less effective than the Flat-Field Correction method.

3.3.5 Flattening Technique

After acquiring the images, this method is used to investigate the effect of correcting the wide-angle lens distortion on detection accuracy. Then, compare the performance of the model with and without applying a flattening technique.

The first technique that been used is fisheyes image correction (Gai et al., 2024). Several parameters related to the camera setup and image dimensions are defined. First, the image dimensions are specified with a width of 640 pixels and a height of 640 pixels. These values represent the dimensions of the input image to be corrected for fisheye distortion. Next, the focal lengths along the x and y axes are set to 640 x 600 pixels, respectively. These values determine the apparent magnification of the image due to the fisheye lens. The principal point, representing the optical centre of the image, is calculated as half of the width and height, placing it at the centre of the image. Finally, the camera matrix K is constructed using these parameters. This matrix encapsulates intrinsic camera properties, such as the focal lengths and principal point, necessary for distortion correction algorithms. It is a 3x3 matrix where the upper-left 2x2 submatrix represents the focal lengths and the centre coordinates, and the last row represents the perspective projection (Aini et al., 2025).

The second technique used in this study is stretch where it will stretch the image and compressed it to 640 x 640 pixels. This technique will make the image stretched to make the object of interest appear bigger.

The flow for both Fisheyes Image Correction and stretch were the same except how it processes the image. Fisheyes Image Correction reduces the fisheye effect from the wide-angle lens camera while stretch doesn’t reduce the fisheye effect instead stretch the image to fill the 640 x 640 pixels.

3.3.6 Reflection Reduction

Two techniques were selected in this study for reducing the reflection and glare in the images. 2D Image Flat Field Correction, this technique utilizes a flat-field image (often captured under uniform illumination conditions) as a reference (Wen et al., 2024). The captured image with glare/reflection is then divided by the flat-field image, pixel-wise. This process aims to remove the non-uniformity in illumination caused by the lens and other optical elements, effectively reducing the intensity variations associated with glare and reflection. The intensity was set to 15 for this study. The illumination can balance out the glare or the reflection present in the images (Aini et al., 2025).

Non-Local Mean Filter search mechanism can potentially adapt to varying degrees and patterns of glare/reflection within an image (Heo et al., 2020). By searching for the most similar patches, it can adjust its behaviour based on the specific characteristics of the glare in different image regions. The value was set to 3, 3, 7 where first value represents the size of the pixel neighbourhood used to compute the weights. A larger value will result in a smoother image but may lose some finer details. The second value represents the size of the pixel neighbourhood used to search for similar patches. Like the previous parameter, a larger value will result in a smoother image but may lose some finer details. The last value represents the strength of the filter. A higher value will result in more aggressive denoising but may also blur the image. For NLM, it reduces the glare or the reflection by patching the glared or reflected area of the images. Then, compare the model's performance with using 2D image flat field correction and non-local mean filter. Note that augmentation is also being used to further refine the detection (Aini et al., 2025).

3.3.7 Image Augmentation

For image augmentation, this study used Roboflow, an online platform that helps create and modify datasets for training. The goal was to examine how augmentation techniques like rotation and flipping affect model robustness and generalization (Mumuni, 2022). The augmented images were tested both with and without the previously mentioned glare-reduction techniques. A spatial-level transformation approach was applied, meaning both the images and their bounding boxes were altered. While this makes the coding process a bit more complex, it's proven to be more effective in boosting object detection performance. The transformations used included, random flip (horizontal and vertical), random brightness adjustment (-10% to +10%) and random exposure adjustment (-10% to +10%). Through augmentation, the dataset expanded from just 120 images to 600. Out of these, 480 were used for training, 60 for validation, and 60 for testing. Table 3.2 shows the dataset description and augmentation used in this study.

Table 3.2
Data Acquisition

Dataset description		**Augmentation**
Images taken	300	300
Images chosen	150	150 x 4 = 600
Training images	120	480
Testing images	15	60
Validation images	15	60

3.3.8 YOLO Object Detection

The dataset will undergo a detection process by using YOLOv7. This step is crucial for the overall accuracy of the object detection system as it determines which objects are present in the image and their locations especially when determining the moulted and non-moulted crab. In this machine learning algorithm, it will find the pattern in the training datasets and build a model that captures these patterns. Roboflow will be used to annotate and highlight the object in the image. The dataset was divided to 80:20 ratios. After undergoing all the previous steps, the custom dataset will go through model training. The input image will be processed in the training model to generate predictions for object locations and class probabilities.

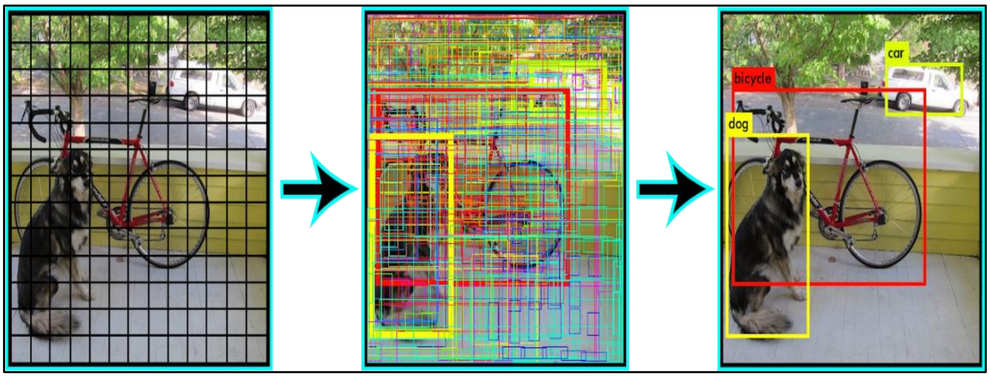


Figure 3.8 YOLO bounding box

As shown in Figure 3.8, the model will divide the image into a grid of cells and make predictions for each cell. Based on the predictions, generate bounding boxes around the objects in the image. The bounding boxes will be drawn using the centre coordinates, width, and height of the objects. The process is shown in Figure 3.9. It was only one class used in the dataset. This is because this study is only focused on detecting the crabs and to improve the accuracy of the detection model based on the open greenhouse farm environment.

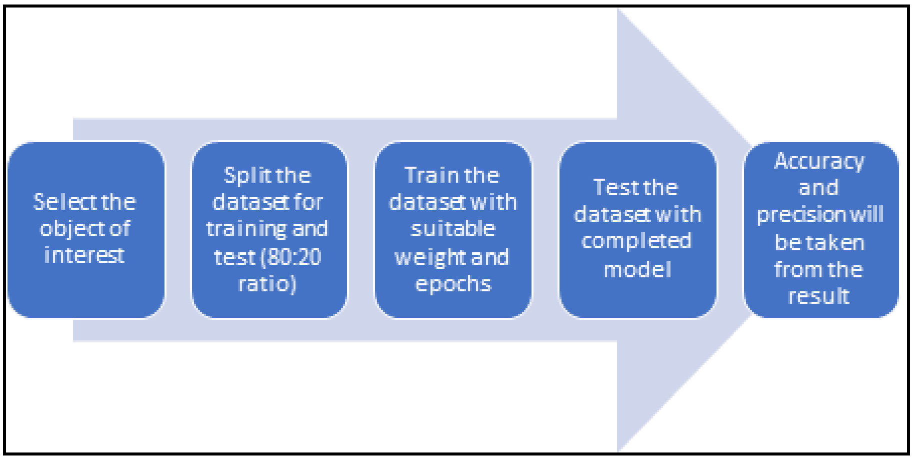


Figure 3.9 Detection process for YOLOv7

This study used the YOLOv7 model for object detection. During testing, tweaking the training settings led to improved results. After pre-processing, the dataset was split into an 80:10:10 ratio with 80% for training, 10% for validation, and 10% for testing (Hu et al., 2021; Nimitkul et al., 2022). Using 80% of the data for training gives the model plenty of examples to learn patterns and relationships, which is key for building a strong, generalizable model. The 10% validation set is used during training for tasks like hyperparameter tuning and monitoring performance on unseen data, helping to avoid overfitting. Finally, the last 10% is kept as a test set, providing an unbiased measure of how well the model performs in real-world scenarios. The

80:10:10 split is widely considered a standard practice in machine learning, as it not only balances training and evaluation but also makes it easier for researchers to compare and reproduce results across studies. Training a custom YOLO model also requires careful configuration of different parameters to ensure the model learns effectively and performs at its best.

Start with --workers 1 parameter, which dictates the number of CPU workers utilized during training. By setting it to 1, the command ensures that only one CPU core is engaged in data loading and preprocessing tasks. This approach helps mitigate potential resource contention and ensures smooth execution of training operations without overwhelming system resources.

Furthermore, the --device 0 parameter specifies the GPU device used for training. In this context, selecting 0 designates the first GPU device for training purposes. Leveraging GPUs accelerates the training process by harnessing their parallel processing capabilities, thereby expediting model convergence and enhancing overall training efficiency.

The parameter --batch-size 7 defines the batch size for training, determining the number of samples processed simultaneously during each training iteration. The --epochs 50 parameter specifies the number of training epochs, representing complete passes through the entire training dataset. Training over multiple epochs allows the model to iteratively refine its parameters and learn robust representations from the data. Initially, this study used 10 batch and 100 epochs. But when the model evaluated with test images, the results are not what as expected. It is because of overfitting and the processing capability of the hardware used. Then, the training was cut down to 7 batch 50 epochs. With a batch size of 7, the model processes seven samples per iteration, striking a balance between computational efficiency and memory utilization.

The --img 640 640 parameter defines the input image size for training, indicating that input images are resized to a square shape with dimensions of 640x640 pixels. Consistent image sizes facilitate uniform processing and enable the model to learn representations effectively across the dataset.

Additionally, the configuration files such as 'data/custom_data.yaml', 'data/hyp.scratch.custom.yaml', and 'cfg/training/yolov7custom.yaml' contain essential metadata, hyperparameters, and model configurations required for training the custom which in this study change how many classes trained (1 class) and what is the name of the class (Crab).

Lastly, the --weights yolov7.pt parameter points to the pre-trained YOLOv7 weights file (yolov7.pt), which serves as the initialization parameters for the custom model. Incorporating pre-trained weights expedites training and enhances convergence by leveraging previously learned features and representations. Then, after the training, the model will create a new custom weight (yolov7_custom.pt). The custom weight will then use in the testing.

In YOLOv7 there are several key parameters to achieve accurate object detection. These parameters fine-tune the model's inference process, where it analyses new data to identify objects of interest, which in this study the crabs.

The root for object detection is in the pre-trained weights file, designated by "--weights" (e.g., yolov7_custom.pt). This file contains the model's expertise, accumulated through prior training on a vast dataset. The specific weights file determines the classes of objects YOLOv7 can recognize within images or videos.

Another critical parameter is the confidence threshold, set using "--conf". This value establishes the minimum level of certainty required by the model to classify an object. For example, "--conf 0.3" sets a threshold of 30%. Objects with a detection confidence below this value are disregarded. This parameter offers a balance between precision (correctly identified objects) and recall (detecting all possible objects). A higher confidence threshold ensures a higher accuracy rate but might lead to missing some valid detections.

Furthermore, "--img-size" defines the dimensions the model expects for the input image. Resizing the input image to this specified size ensures compatibility with the model's internal architecture. The choice of image size can impact processing speed and accuracy. In this study, the image size used was 640 x 640 resolution. A larger image size might capture more details but requires more computational resources.

The "--source" parameter specifies the data source for object detection. This can be a single image file, a video file, or a directory containing multiple images or videos. This parameter dictates the data the model will analyse to detect objects which for this study are the crab images.

Finally, the inclusion of "--save.txt" enables the model to save the detection results in a text file. This file serves as a record of the identified objects and their corresponding attributes, facilitating further analysis or visualization of the results.

Table 3.3
YOLOv7 Parameters

Parameter	Type	Value Used	Description
--workers	Training	1	Number of CPU cores for data loading.
--device	Training	0	Specifies the GPU device for training.
--batch-size	Training	7	Number of images processed per iteration. Reduced from 10 to 7 to fit hardware limits.
--epochs	Training	50	Total passes through the dataset. Reduced from 100 to 50 to prevent overfitting.
--img	Both	640 x 640	Resolution of the images for uniform processing.
--weights	Both	yolov7.pt (initial)	Pre-trained weights used for Transfer Learning.
--conf	Inference	0.3 (30%)	Minimum certainty required to "confirm" a crab detection.
--source	Inference	Variable	Path to the test images or video being analysed.
--save-txt	Inference	Enabled	Saves detection coordinates and classes into text files for analysis.

YOLOv7 was chosen for this study because of its robustness and there were handful of resources available that can act as a guideline for this study. Table 3.3 shows the parameters used in this study. The parameters was based on previous studies by C. Y. Wang et al. (2022). No modification in YOLOv7 architecture during the study.

3.3.9 Crab Counting

Counting in object detection involves the process of quantifying specific objects within an image or video frame using computer vision techniques. The "ground truth" represents manually annotated instances of objects within the data, serving as a benchmark for evaluating the performance of object detection algorithms (Guo et al., 2022). Conversely, "crab detected" refers to instances where the algorithm has identified crabs within the same data. By comparing the ground truth with the crab detections, thus can assess the accuracy and effectiveness of the detection model (Iftikhar et al., 2024).

The counting works well when the position of the container is predetermined section by section. The chances of the cart to skip containers is low. The cart will move forward to endpoint and will stop at dedicated section to take pictures for every 30 seconds. Then it will go back to starting point and still taking pictures. The order of the position will be reversed when the cart move back to the starting point. Every time the cart taking picture, the data will be sent to the pc to start the real time detection. The model will count the crabs for each section. The coding for the microcontroller to control the cart is in appendices.

3.4 Phase 3: Evaluation

3.4.1 Output

The output will show the bounding boxes on the input image, along with the class labels and confidence scores. The algorithm detects the moulted crab as two crabs and this will be fine-tuned to label it as moulted crab. Figure 3.9 shows the preliminary testing of the YOLOv7 object detection model in the lab. Figure 3.10 shows the preliminary testing with images taken from the farm. Meanwhile Figure 3.11 shows the moulting detection in previous study using YOLOv4 (Hu et al., 2021). The process of object detection involves three main stages: detection, recognition, and identification. Detection is the initial stage where an object is identified in an image. Recognition is the process of determining the category of the detected object, while identification involves assigning a unique identity to the detected object. Later, in this study, the detection algorithm will detect the moulting crab by using YOLOv7.

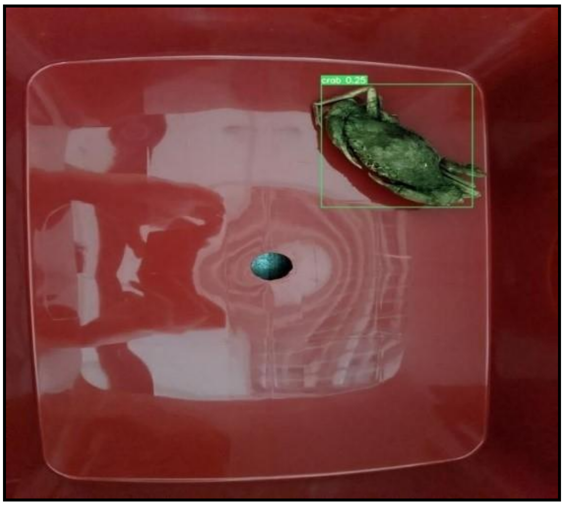


Figure 3.10 Preliminary testing for YOLOv7 in lab

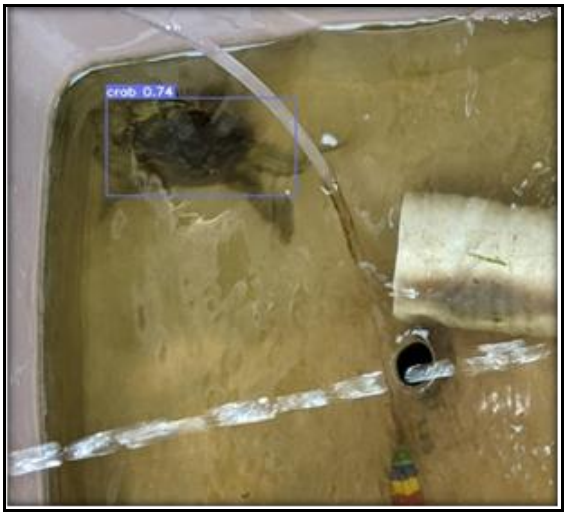


Figure 3.11 Preliminary testing for YOLOv7 using image from crab farm

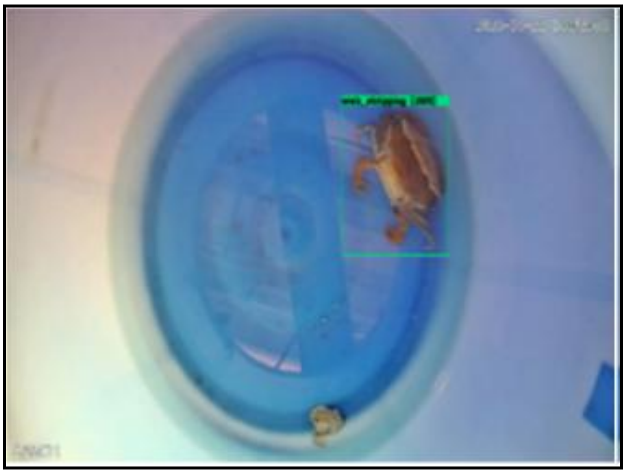


Figure 3.12 Moulting detection in previous study using YOLOv4 (Hu et al., 2021)

3.4.2 Performance Metrics

After testing the images with the stated techniques above, using the YOLOv7 object detection model, each result was evaluated to assess the model’s performance. Recall, Precision, and F1 Score were used to evaluate the performance of detection after and before the images are enhanced. The recall is for how many crabs that are detected. A high recall indicates that the model can effectively detect most crabs present in the images without missing many. Precision is how accurate the detection is to detect the crabs. In the context of crab detection, precision indicates how many of the detected instances are crabs, preventing false positives. A high precision suggests that the model is not labelling too many non-crab objects as crabs. F1 Score is the overall performance for the detection. It provides a balanced measure of a model's performance, considering both false positives (precision) and false negatives (recall). A high F1 score indicates that the model has both high precision and high recall, making it a suitable metric for situations where there is an imbalance between positive and negative instances or where both precision and recall is equally important. The formulas are shown below.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Figure 3.13 Formula for Precision, Recall, and F1-Score.

CHAPTER 4
RESULT AND DISCUSSION

4.1 Discussion

Pre-requisite for this study is to setup an IoT system for monitoring the crabs. In which include the installation of the custom cart mentioned in previous chapter. The custom cart was hanged on the railing that were attached to the pillar of the greenhouse. Two cameras were mounted on the custom cart. The cameras and the microcontroller that controlled the custom cart were connected to a router with internet connectivity for data gathering and monitoring. The coding and example of the web application for this study is in the appendices.

The first part of the experiment focused on improving the detection of crab images that appear distorted when captured by a wide-angle camera especially when the crab is positioned near the edges of the frame. To address this, a flattening method was applied, and the results were promising. Testing with crabs placed in different edge positions showed consistent improvements. In Table 2, Flat + Augmentation, the approach performs well across all metrics, achieving high recall, precision, and F1 scores while detecting 78.57% of the crabs. It shows an improvement of approximately 23.21% compared to the No Processing approach in terms of the percentage of detected crabs. Flattening works by converting a multi-dimensional array (such as a 2D image) into a one-dimensional array while preserving spatial details (Yang et al., 2020). This helps the model extract features more effectively. Augmentation techniques such as rotation, scaling, translation, and flipping can further enhance feature extraction by presenting the model with a diverse set of variations of the same image, helping it learn robust features.

For the second part of the experiment, we found that glare or reflection can affect the detection of crabs in the image. To solve this problem, we used 2D-Image Flat Field Correction (Kim & Baek, 2018) and Non-Local Mean Filter (Heo et al., 2020). Based on the result in Table 3, the 2D-Image Flat Field Correction method can tackle this problem better and show more improvement than the Non-Local Mean Filter. For the result in Table 3, 2D Image Flat Field Correction + Augmentation, outperformed the No Processing approach in crab detection. The percentage difference in crab detection between 2D Image Flat Field Correction + Augmentation and No Processing is approximately 25.64%. 2D image flat field correction improves detection models by reducing background noise, normalizing intensity levels, enhancing feature extraction, and improving generalization capabilities (Kokka et al., 2019). By providing cleaner and more consistent input data, flat field correction helps detection models achieve better performance and accuracy in identifying objects of interest within images (Aini et al., 2025). Once again Augmentation shows it is a crucial approach to further improve the detection.

In the third part of this experiment, we combine the flattening and 2D Image Flat Field Correction methods to test whether the result will be improved or not. The recall rate for this approach is the highest compared to other experiments done before based on the recorded results.

Table 4.1
Combined Method

Experiment	**Number of crabs detected**	**Percentage of crabs detected (%)**
No Processing	140	64.8148
Fisheyes Image Correction + 2D Image Flat Field Correction	174	80.5556
Fisheyes Image Correction + 2D Image Flat Field Correction + Augmentation	195	90.2778

It's important to highlight that data augmentation played a key role in this study. In machine learning, augmentation is used to artificially expand the training dataset by applying different transformations to existing images (Mumuni, 2022; Shorten & Khoshgoftaar, 2019). The detection model was tested in multiple experiments both with and without augmentation. The results showed that augmentation significantly improved the model's performance by helping it generalize better and recognize a wider range of patterns. By increasing the diversity of training examples, the model was able to handle variations in the appearance of crabs, whether they were located at the edges or the centre of the image. This led to stronger, more reliable predictions on unseen data.

When comparing results, the difference between using no augmentation and augmentation was clear. The experiments that combined flattened image data with augmentation consistently outperformed all others in detecting edge-positioned crabs, based on the F1 Score. This approach also achieved higher recall and precision than the other setups. These enhancements proved especially valuable for correcting distortions caused by wide-angle cameras, allowing the model to detect crabs more accurately.

On the other hand, the detection performance between the two scenarios no augmentation and augmented, where 2D Image Flat-Field Correction, outperforms every experimentation based on the F1 Score. It possesses high recall value and a little trade-off in precision. The enhancements play a crucial role in optimizing the model's sensitivity to darker regions, resulting in a more balanced overall performance (Kim & Baek, 2018). The capability of the 2D Image Flat-Field Correction excels in detecting images in darker areas and glare signifies a practical advantage in real-world scenarios where lighting conditions may vary (Kokka et al., 2019). This adaptability is reflected in the F1 Score, emphasizing the importance of considering the specific challenges posed by the environment in which the model will be deployed (Aini et al., 2025).

In contrast, when comparing the no augmentation scenarios with 2D Image Flat-Field Correction, this method outperformed all other experiments based on the F1 Score. It achieved a high recall, though with a slight trade-off in precision. These improvements were important for enhancing the model's sensitivity to darker regions, leading to a more balanced overall performance (Kim & Baek, 2018). One of the key strengths of 2D Image Flat-Field Correction is its ability to handle challenges like low-light areas and glare conditions that are very common in real-world environments where lighting is not always ideal (Kokka et al., 2019).

YOLOv7 (You Only Look Once) was chosen for crab detection in this study because of its strong mix of real-time speed, accuracy, and adaptability (C. Y. Wang et al., 2022). Its ability to quickly process large sets of images makes it highly efficient for timely crab identification, something especially valuable in applications like environmental monitoring. Since YOLOv7 uses a single-pass detection approach, it scans the entire image in one go, which boosts both speed and efficiency. On top of that, it delivers high accuracy, adapts well to changes in scale and aspect ratio, and is relatively easy to implement. These strengths make it reliable for detecting crabs under different environmental conditions. In short, the combination of speed, precision, and flexibility makes YOLOv7 a powerful and practical choice for crab detection in this study.

That said, YOLOv7 does have some limitations in this study, particularly when it comes to detecting small or partially hidden crabs (Yu et al., 2023). While its single-pass approach is fast, it can struggle to accurately pick out tiny objects in cluttered backgrounds or when crabs appear at different scales. Another challenge is occlusion—

if a crab is partially covered by other objects or environmental elements, YOLOv7 may miss it entirely or produce inaccurate bounding boxes (Sun et al., 2023). These issues can lower overall detection performance, which is especially problematic in applications where precise localization is critical, such as ecological research or environmental monitoring (Aini et al., 2025).

Among all the reviewed pre-processing techniques in the Literature Review, the most significant to this research is 2D-image flat-field correction, which addresses uneven illumination caused by variable lighting in greenhouse aquaculture environments. This technique normalizes intensity variations across the image, reducing glare and balancing lighting, thereby enhancing the quality and consistency of the dataset for accurate detection. In addition, fisheye flattening plays a crucial role in correcting geometric distortions caused by wide-angle lenses, ensuring that crabs appear in their true proportions regardless of their position in the frame (Aini et al., 2025). Together, these techniques form a robust pre-processing pipeline, significantly improving the quality of the dataset and enhancing the performance of detection algorithms.

This research explicitly contributes in several ways. First, a minor dataset was developed, featuring images captured under real-world aquaculture conditions, such as glare, reflections, and wide-angle distortions. This dataset serves as a baseline for the study. Second, the research transformed the raw dataset into an enhanced dataset by applying the pre-processing techniques of 2D-image flat-field correction and fisheye flattening, effectively mitigating environmental challenges. The impact of these enhancements is evaluated, where the pre-processed data is shown to significantly improve the accuracy of detection algorithms like YOLO. Furthermore, the research demonstrates how enhanced pre-processing enables real-time detection of moulting crabs, addressing critical moments in aquaculture operations.

The thesis provides a detailed breakdown of these contributions, where the dataset and pre-processing pipeline are described, where experimental results validate the approach. The limitations of the study, such as the dependency on specific environmental conditions and challenges in generalizing the approach to other setups. Assumptions, including that each container holds one crab for accurate counting and detection to easily locate and evaluate the contributions, providing a clear narrative of how enhanced pre-processing techniques significantly improve soft-shell crab detection in greenhouse aquaculture environments.

4.2 Crab Detection

This study will present and compare a few experimentations with the same dataset to determine which experiment that detect most of the crabs in an image. This task poses several challenges, including the presence of glare and crab images that are distorted because of wide-angle camera.

To train a custom YOLOv7 model for image recognition, start by preparing the dataset, which consists of 300 images split into training, validation, and test sets using an 80:10:10 ratio. This results in 240 images for training, 30 for validation, and 30 for testing.

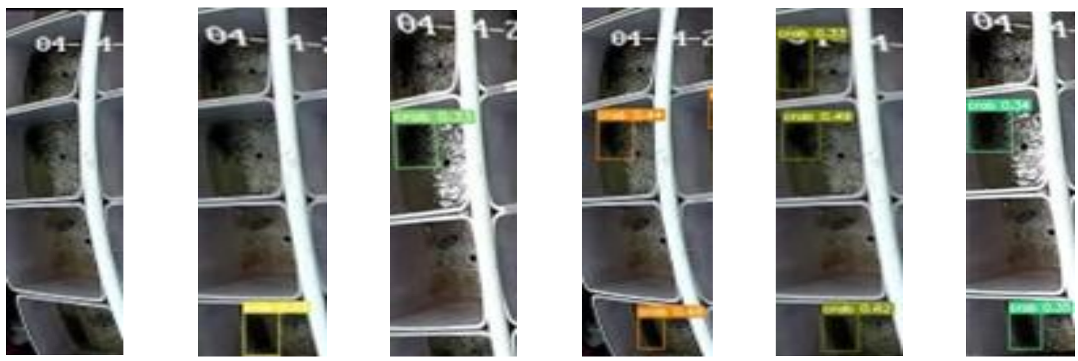
One data loading worker is specified, and the first GPU is used for training, setting a batch size of 7 images per batch to balance memory usage and training speed. The model is trained over 50 epochs with images resized to 640x640 pixels. The dataset configuration, including paths to training, validation, and test sets, is provided in the custom_data.yaml file, while training hyperparameters are defined in hyp.scratch.custom.yaml. The model architecture is detailed in yolov7custom.yaml, tailored for our specific task. The training named as yolov7-custom for easy identification and initialize the model with pre-trained weights from yolov7.pt. After training, the model's performance is evaluated on the test set using metrics such as precision, recall, F1 score, and mean Average Precision (mAP) to ensure it generalizes well to new data, ensuring the model is effectively trained and fine-tuned for image recognition.

The confidence threshold is set to 0.3 using the --conf 0.3 parameter, ensuring that only detections with a confidence score of 30% or higher are considered valid, thus filtering out low-confidence detections. The --img 640 parameter specifies that all input images will be resized to 640x640 pixels, matching the dimensions used during training for consistency. The --source imagesfile parameter defines the directory containing the test images to be processed. Detection results are saved in text format using the --save-txt parameter, which includes bounding box coordinates, class labels, and confidence scores for each detected object. The command is executed in the terminal, and the model processes each image in the specified directory, generating predictions based on the input parameters. The saved detection results are reviewed and analysed to assess the model's performance both visually and quantitatively. Metrics such as precision, recall and F1 score are calculated to evaluate the model's accuracy. This comprehensive evaluation process ensures that the YOLOv7 model accurately detects and classifies objects in new, unseen images, confirming its readiness for deployment or identifying areas for further refinement.

Table 4.2 shows the result of detection of the crab at the edge in the image. The flattening method shows an improvement in recall but slightly off in precision. After including the augmentation with the flattening method, the result shows that the detection is significantly improved.

Table 4.2
Result for Detection of The Crab At The Edge of The Image

Experiment	**Recall**	**Precision**	**F1 Score**	**Number of crabs detected**	**Percentage of crabs detected (%)**
No Processing	0.553	1	0.712	31	55.35
Fisheyes Image Correction	0.750	0.976	0.848	42	75.00
Stretch	0.744	0.943	0.832	40	71.43
Data Augmentation	0.678	1	0.808	38	67.85
Fisheyes Image Correction + Data Augmentation	0.785	1	0.880	44	78.57
Stretch + Data Augmentation	0.763	0.982	0.859	43	76.79



(a) (b) (c) (d) (e) (f)

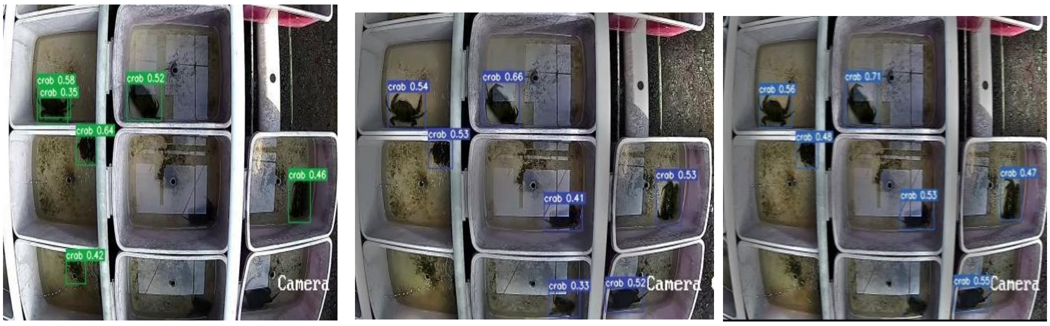
Figure 4.1 Detection of crab with different image pre-processing to focus on the crab at the edge of the image. (a) No Processing, (b) Fisheyes Image Correction, (c) Stretch, (d) Augmentation, (e) Fisheyes Image Correction + Augmentation, (f) Stretch + Augmentation

As we can observed in Figure 4.1, image in (e) Fisheyes Image Correction + Data Augmentation, detect the most crab in the image. For (b) Fisheyes Image Correction and Stretch, only detect one crab out of three crabs in the image, while (d) Data Augmentation and (f) Stretch + Data Augmentation, shows only two crabs out of three. The image in (a) No Processing, does not detect any crabs at all.

Table 4.3 shows the result for detection of the crab in the middle of the image. 2D Image Flat Field Correction shows the improvement in recall and precision compared to No Processing. Between 2D Image Flat Field Correction and Non-Local Mean Filter, 2D Image Flat Field Correction performs better in Recall but Non-Local Mean Filter slightly better in Precision shows that 2D Image Flat Field Correction have less false positive than Non-Local Mean Filter and Non-Local Mean Filter have a bit more precise than 2D Image Flat Field Correction when detecting the crabs. For F1 Score, 2D Image Flat Field Correction perform better overall score than Non-Local Mean Filter. After including the augmentation with 2D Image Flat Field Correction, the overall result which is F1 Score shows that the detection is significantly improved.

Table 4.3
Result for Detection of The Crab In The Middle of The Image

Experiment	**Recall**	**Precision**	**F1 Score**	**Number of crabs detected**	**Percentage of crabs detected (%)**
No Processing	0.681	0.956	0.795	109	68.12
2D Image Flat Field Correction	0.781	0.947	0.856	125	78.12
Non-Local Mean Filter	0.743	0.967	0.841	119	74.37
Augmentation	0.750	0.967	0.845	120	75.00
2D Image Flat Field Correction + Augmentation	0.856	0.944	0.898	137	85.62
Non-Local Mean Filter + Augmentation	0.831	0.963	0.892	133	83.12



a) b) c)

Figure 4.2 Detection of crab with different image pre-processing to focus on the crab at the middle of the image. (a) No Processing, (b) 2D Image Flat Field Correction, (c)Non-Local Mean Filter

Figure 4.2 shows, the image in (b) 2D Image Flat Field Correction and (c) Non-Local Mean Filter, detect the most crab in the image. For (a) No Processing, it detects but not all the crabs in the image, there is also an overlap in one of the detected crabs.

a) b) c)

Figure 4.3 Detection of crab with different image pre-processing and augmentation to focus on the crab at the middle of the image. (a) Augmentation, (b) 2D Image Flat Field Correction + Augmentation, (c) Non-Local Mean Filter + Augmentation.

In Figure 4.3, this experiment is to test the effect of augmentation. Augmentation can improve the performance of object detection when paired with suitable processing. As we can see in (b) 2D Image Flat Field Correction + Augmentation detects all the crabs in the image while (a) Augmentation, and (c) Non-Local Mean Filter only missed one of the crabs in the image.

Table 4.3 shows the result of the detection of the crab in the image. When combining Flattening with 2D Image Flat Field Correction, the results show that it improved compared to No Processing. The result significantly improved when including the augmentation to Flattening and 2D Image Flat Field Correction.

Table 4.4
Result for Detection for all the crabs in the image

Experiment	Recall	Precision	F1 Score	Number of crabs detected	Percentage of crabs detected (%)
No Processing	0.648	0.965	0.775	140	64.81
Fisheyes Image Correction + 2D Image Flat Field Correction	0.805	0.956	0.874	174	80.56
Fisheyes Image Correction + 2D Image Flat Field Correction + Augmentation	0.902	0.955	0.928	195	90.23
Fisheye Flattening Method + 2D-Image Flat Field Correction + Augmentation (5-fold cross validation)	0.895	0.952	0.923	193	89.52

a) b) c)

Figure 4.4 Detection of crab for whole image captured with different image pre-processing. (a) No Processing, (b) Fisheyes Image Correction + 2D Image Flat Field Correction, (c) Fisheyes Image Correction + 2D Image Flat Field Correction + Augmentation

Observing Figure 4.4, we can see that in (c) Fisheyes Image Correction + 2D Image Flat Field Correction + Augmentation detects all the crabs present in the image. Augmentation is an important part of the improvement.

In the initial phase of evaluation, the model was tested using a limited dataset comprising only 10% of the total images (60 images). To address this limitation and enhance the reliability of the findings, a 5-fold cross-validation approach (k = 5) was employed. This method increased the volume of testing data to 600 images, which were processed using a combination of Fisheye Flattening and 2D Image Flat Field Correction with data augmentation. The cross-validation results, presented in Table 4.3, demonstrate a strong consistency with the earlier 10% split evaluation. For instance, the recall achieved through 5-fold cross-validation was 0.895 compared to 0.902 from the 10% split; precision was 0.952 versus 0.956; the F1 Score was 0.923 versus 0.929; and the crab detection rate was 89.52% versus 90.32%. This alignment between the two validation methods reinforces the robustness of the proposed model and supports its feasibility and generalisability for real-world soft-shell crab detection applications.

4.3 Counting Performance

This method was to validate the efficacy of the method designed for detecting crabs. For this purpose, images taken from the crab farm were gathered. Three distinct scenarios were observed:

Scenario 1: The wide-angled camera lens distorted the images of the crab, especially at the edge.

Scenario 2: The glare/reflection that is present in the image.

Scenario 3: The overall results for the detection.

For crab detection, precision, recall, and F1 would be particularly important metrics. The exact performance numbers would depend on the specific implementation, the dataset used for training and evaluation, and the requirements of the application. The ground truth data were used to verify the results from manually counting the data. The false-positive count indicates the number of false detections. False-negative to count the missed detections. Accuracy is used as the basis for evaluation which higher values indicate closer to the actual results. When large-scale crab counts are needed, the method used can significantly reduce the counting time. Therefore, the proposed method can effectively reduce manual labour when the yield of the crabs needs to be assessed. These scenarios will be discussed in the next section (Aini et al., 2025).

4.4 Moulting Detection

In this research, an IoT system was employed to integrate cameras and image detection for real-time, automatic detection of moulting. The system captures images of each crab in individual containers, processes the images to detect signs of moulting, and highlights the crabs that have moulted (Figure 4.5). If there the system detects two crabs in one container, it will detect as moulted. This capability to simultaneously monitor multiple containers without the need for direct human intervention represents a significant advancement over traditional, labour-intensive methods.

This study aligns with previous research that employed the YOLOv4 model for object detection (Hu et al., 2021), specifically in a review where an underwater camera with night vision capabilities was used to monitor aquatic environments. In contrast, my study utilises a more conventional approach by deploying standard outdoor CCTV cameras without specialised night vision features. The key innovation in this study lies in the pre-processing and fine-tuning of images to enhance detection accuracy, compensating for the lack of advanced camera technology. This strategic focus on software-driven image enhancement allows for the effective detection of soft-shell crabs despite using less sophisticated hardware.

In terms of hardware and overall system design, this study approach offers significant advantages. The use of regular CCTV cameras, combined with image processing techniques, results in a system that is not only effective but also cost-efficient. Unlike the setup with underwater cameras and night vision, which can be expensive and complex to implement, method in this study emphasise the use of affordable, readily available hardware components. The emphasis on coding and image processing rather than high-end hardware leads to a reduction in costs, making the system more accessible and easier to deploy in various aquaculture settings. This cost-effectiveness, combined with the robust performance of the detection system, justifies the better approach, especially for applications where budget constraints are a critical consideration.

Figure 4.5 Moulted soft-shell crab detected with the IoT setup and YOLOv7 Object Detection

The accuracy of moulting detection is highly dependent on the quality of the images captured by the cameras and the performance of the image detection algorithms. Poor lighting conditions, reflections on the water surface, and the similarity in colour between the crab and the container with murky water could potentially lead to false detections or missed moulting events. Furthermore, the effectiveness of the system can be affected by the crab's positioning within the container and the camera's angle. These variables underscore the significance of optimizing camera placement and ensuring consistent lighting to improve detection accuracy.

CHAPTER 5
CONCLUSION

5.1 Conclusion

This study successfully met all its objectives based on the results obtained. The experiments were designed to tackle the challenges of detecting crabs in images affected by glare and distortion from wide-angle camera lenses. By testing different techniques, it became clear that combining flattening with 2D Image Flat Field Correction led to a significant improvement in detection rates. These methods effectively reduced distortion and glare, resulting in higher recall, precision, and F1 scores compared to other approaches. Another key factor was data augmentation, which played an important role in boosting the model's performance. By generating more diverse training examples, augmentation improved the model's ability to generalize, making it more reliable in detecting crabs under varying conditions.

While YOLO proved to be a suitable choice for crab detection due to its real-time processing capabilities and accuracy, it also demonstrated limitations in detecting small or partially occluded crabs. Despite these weaknesses, the overall findings highlight the importance of employing appropriate pre-processing techniques and augmentation strategies to enhance the performance of object detection models in challenging environmental conditions.

This study underscores the significance of considering specific challenges and selecting suitable methodologies to achieve accurate and reliable crab detection in image datasets. From aquaculture perspective, this helps crab farmers to increase their production yield and profits without need for constant human on-site supervision. Thus, motivate the active involvement of the public to venture into this industry at large-scale production. The high production would upgrade the stocking of trade, expand the exports, and directly increase the soft-shell crab production for the domestic and international markets.

As a result, it also guarantees the Food Supply to build a resilient key new sector for Commodity 2.0, in line with Dasar Agromakanan Negara 2011-2030 and shared Prosperity Vision 2030 (WKB 2030) and the SDG 14 (Life Below Water) program.

5.2 Future work

For future work, it is recommended to delve deeper into optimizing the model's adaptability and performance under varying environmental conditions. Specifically, fine-tuning the parameters related to enhancements could be explored to achieve an even more precise balance between precision and recall. Conducting a thorough analysis of how distinct levels of enhancement impact detection performance and understanding the optimal thresholds for these enhancements would contribute to refining the model for a wider range of scenarios. Additionally, investigating the impact of modifications on different object classes or image characteristics may provide insights into the generalizability of the model. Future research could explore using more advanced image processing methods and improved model architectures to make the detection system even more robust. In addition, building a larger and more diverse dataset covering different types of images and environmental conditions would allow for a more thorough evaluation of the model's performance. Finally, deploying the model in real-world scenarios and assessing its performance in practical applications would offer valuable feedback for further improvements and optimizations.

REFERENCES

Abdel-Hamid, O., Mohamed, A. r., Jiang, H., Deng, L., Penn, G., & Yu, D. (2014). Convolutional Neural Networks for Speech Recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *22*(10), 1533-1545. https://doi.org/10.1109/TASLP.2014.2339736

Abebe, M. A., Booth, A., Kervec, J., Pouli, T., & Larabi, M.-C. (2018). Towards an automatic correction of over-exposure in photographs: Application to tone-mapping. *Computer Vision and Image Understanding*, *168*, 3-20. https://doi.org/https://doi.org/10.1016/j.cviu.2017.05.011

Agrawal, A., & Raskar, R. (2009, 20-25 June 2009). Optimal single image capture for motion deblurring. 2009 IEEE Conference on Computer Vision and Pattern Recognition,

Aini, M. A. B., Leong, S. H., Yong, Y. T., Lee, B. Y., Zhao, X., Manaf, S. R., Abdullah, F., & Khong, H. Y. (2025). Image Enhancement for Detection of Underwater Moulted Crabs in Greenhouse Soft-shell Crab Farming Using Deep Learning. *Smart Agricultural Technology*, 101223. https://doi.org/https://doi.org/10.1016/j.atech.2025.101223

Akhter, R., & Sofi, S. A. (2022). Precision agriculture using IoT data analytics and machine learning. *Journal of King Saud University - Computer and Information Sciences*, *34*(8, Part B), 5602-5618. https://doi.org/https://doi.org/10.1016/j.jksuci.2021.05.013

Aladdin, A. M., Hasan, D. O., Mohammed-Taha, S. R., Mustafa, A. I., & Badawi, S. (2026). Integrating wearable sensors in pandemic healthcare: a systematic review of IoT-based detection and protection technologies. *Measurement*, *262*, 119991. https://doi.org/https://doi.org/10.1016/j.measurement.2025.119991

Almarinez, J., & Hernandez, A. (2019, 15-17 Aug. 2019). Classifying Mangrove Crub Images for Growth Stages Detection and Monitoring. 2019 International Conference on Sensing, Diagnostics, Prognostics, and Control (SDPC),

Alymani, M. (2025). The Impact of the Internet of Things (IoT) on Enhancing Smart Cities: A Literature Review. In (pp. 336-343). https://doi.org/10.1007/978-3-032-00444-4_32

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, *8*(1), 53. https://doi.org/10.1186/s40537-021-00444-8

Anozie, U., Pieterson, K., Onyenahazi, O., Chukwuebuka, U., & Ekeocha, P. (2024). Integration of IoT technology in lean manufacturing for real-time supply chain optimization. *International Journal of Science and Research Archive*, *12*, 1948-1957. https://doi.org/10.30574/ijsra.2024.12.2.1498

Baharuddin, R. R., Niswar, M., Ilham, A. A., & Kashihara, S. (2022, 29-30 Jan. 2022). Crab Molting Identification using Machine Learning Classifiers. 2021 International Seminar on Machine Learning, Optimization, and Data Science (ISMODE),

Banerjee, S., Bandyopadhyay, A., Bag, R., & Das, A. (2015, 20-22 Nov. 2015). Sequentially combined mean-median filter for high density salt and pepper noise removal. 2015 IEEE International Conference on Research in Computational

Intelligence and Communication Networks (ICRCICN),
Ben-Lhachemi, N., Benchrifa, M., Nasrdine, S., Mabrouki, J., Slaoui, M., & Azrour, M. a. (2024). Effect of IoT Integration in Agricultural Greenhouses. In J. Mabrouki & A. Mourade (Eds.), *Technical and Technological Solutions Towards a Sustainable Society and Circular Economy* (pp. 435-445). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-56292-1_35
Blessing, M., Olusegun, J., & Owen, J. (2024). Implementing Robust Data Encryption Techniques: Best Practices and Challenges in Developing Effective Security Policies.
Bochkovskiy, A., Wang, C.-Y., & Liao, H.-y. (2020). *YOLOv4: Optimal Speed and Accuracy of Object Detection*.
Boncelet, C. (2009). Chapter 7 - Image Noise Models. In A. Bovik (Ed.), *The Essential Guide to Image Processing* (pp. 143-167). Academic Press. https://doi.org/https://doi.org/10.1016/B978-0-12-374457-9.00007-X
Cervantes, J., Garcia-Lamont, F., Rodríguez-Mazahua, L., & Lopez, A. (2020). A comprehensive survey on support vector machine classification: Applications, challenges and trends. *Neurocomputing*, *408*, 189-215. https://doi.org/https://doi.org/10.1016/j.neucom.2019.10.118
Chaitra, P. G., Deepthi, V., Gautami, S., Suraj, H. M., & Kumar, N. (2020, 2-4 July 2020). Convolutional Neural Network based Working Model of Self Driving Car - a Study. 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC),
Chen, S., Migaud, H., Shi, C., Song, C., Wang, C., Ye, Y., Ren, Z., Wang, H., & Mu, C. (2021). Light intensity impacts on growth, molting and oxidative stress of juvenile mud crab Scylla paramamosain. *Aquaculture*, *545*, 737159. https://doi.org/https://doi.org/10.1016/j.aquaculture.2021.737159
Dalal, N., & Triggs, B. (2005, 20-25 June 2005). Histograms of oriented gradients for human detection. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05),
Deng, G., & Cahill, L. W. (1993). *An adaptive Gaussian filter for noise reduction and edge detection* (Vol. 3). https://doi.org/10.1109/NSSMIC.1993.373563
Fan, L., Zhang, F., Fan, H., & Zhang, C. (2019). Brief review of image denoising techniques. *Visual Computing for Industry, Biomedicine, and Art*, *2*(1), 7. https://doi.org/10.1186/s42492-019-0016-7
Fan, Z., Xie, J.-k., Wang, Z.-y., Liu, P.-C., Qu, S.-j., & Huo, L. (2021). Image Classification Method Based on Improved KNN Algorithm. *Journal of Physics: Conference Series*, *1930*, 012009. https://doi.org/10.1088/1742-6596/1930/1/012009
Fazhan, H., Waiho, K., Ikhwanuddin, M., Shu-Chien, A. C., Fujaya, Y., Wang, Y., Liew, H. J., Chen, C., Abualreesh, M. H., Jaya-Ram, A., & Ahmad-Ideris, A. R. (2022). Limb loss and feeding ability in the juvenile mud crab Scylla olivacea: Implications of limb autotomy for aquaculture practice [Article]. *Applied Animal Behaviour Science*, *247*, Article 105553. https://doi.org/10.1016/j.applanim.2022.105553
Gai, S., Cui, W., & Liang, B. (2024). A fisheye image correction method based on deep learning. *Robotic Intelligence and Automation*, *45*, 70-76. https://doi.org/10.1108/RIA-03-2024-0082
Gaye, B., Zhang, D., & Wulamu, A. (2021). Improvement of Support Vector Machine Algorithm in Big Data Background. *Mathematical Problems in Engineering*, *2021*, 5594899. https://doi.org/10.1155/2021/5594899

Ghadge, P., Tilawat, R., Sand, P., & Jadhav, P. (2021). APPLICATIONS OF COMPUTER VISION IN AGRICULTURE. *International Journal of Engineering Applied Sciences and Technology*, *5*. https://doi.org/10.33564/IJEAST.2021.v05i09.026

Girshick, R. (2015, 7-13 Dec. 2015). Fast R-CNN. 2015 IEEE International Conference on Computer Vision (ICCV),

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014, 23-28 June 2014). Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. 2014 IEEE Conference on Computer Vision and Pattern Recognition,

Guo, G., Wang, H., Bell, D., & Bi, Y. (2004). KNN Model-Based Approach in Classification.

Guo, X., Anisetti, M., Gao, M., & Jeon, G. (2022). Object Counting in Remote Sensing via Triple Attention and Scale-Aware Network. *Remote Sensing*, *14*(24).

Häni, N., Roy, P., & Isler, V. (2018). *Apple Counting using Convolutional Neural Networks*. https://doi.org/10.1109/IROS.2018.8594304

Hassan, M. F., Siaw Lang, W., & Paramesran, R. (2024). An integrated enhancement method to improve image visibility and remove color cast for sand-dust image. *Multimedia Tools and Applications*. https://doi.org/10.1007/s11042-024-19245-1

Heo, Y.-C., Kim, K., & Lee, Y. (2020). Image Denoising Using Non-Local Means (NLM) Approach in Magnetic Resonance (MR) Imaging: A Systematic Review. *Applied Sciences*, *10*(20).

Hou, X., Gong, Y., Liu, B., Sun, K., Liu, J., Xu, B., Duan, J., & Qiu, G. (2018). Learning Based Image Transformation Using Convolutional Neural Networks. *IEEE Access*, *PP*, 1-1. https://doi.org/10.1109/ACCESS.2018.2868733

Hu, K., Shi, C., Gao, G., & Zhu, J. (2021, 10-12 Dec. 2021). A Detection Systems For Molting Scylla Paramamosain Based On YOLO v4. 2021 3rd International Academic Exchange Conference on Science and Technology Innovation (IAECST),

Hu, W. C., Chen, L. B., Hsieh, M. H., & Ting, Y. K. (2023). A Deep-Learning-Based Fast Counting Methodology Using Density Estimation for Counting Shrimp Larvae [Article]. *IEEE Sensors Journal*, *23*(1), 527-535. https://doi.org/10.1109/JSEN.2022.3223334

Iftikhar, M., Neal, M., Hold, N., Gregory Dal Toé, S., & Tiddeman, B. (2024). Detection of Crabs and Lobsters Using a Benchmark Single-Stage Detector and Novel Fisheries Dataset. *Computers*, *13*(5).

Joseph, V. R., & Vakayil, A. (2021). SPlit: An Optimal Method for Data Splitting. *Technometrics*, *64*, 1-23. https://doi.org/10.1080/00401706.2021.1921037

Kalchbrenner, N., & Blunsom, P. (2013). Recurrent continuous translation models. *EMNLP 2013 - 2013 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, *3*, 1700-1709.

Kanagaraj, N., Hicks, D., Goyal, A., Mishra Tiwari, S., & Singh, G. (2021). Deep learning using computer vision in self driving cars for lane and traffic sign detection. *International Journal of System Assurance Engineering and Management*, *12*. https://doi.org/10.1007/s13198-021-01127-6

Kim, D., & Baek, J. (2018, 10-17 Nov. 2018). Comparison of Flat Field Correction Methods for Photon-Counting Spectral CT Images. 2018 IEEE Nuclear Science Symposium and Medical Imaging Conference Proceedings (NSS/MIC),

Kim, J.-W., Ryu, J.-H., & Kim, J.-O. (2020). Deep gradual flash fusion for low-light enhancement. *Journal of Visual Communication and Image Representation*, *72*,

102903. https://doi.org/https://doi.org/10.1016/j.jvcir.2020.102903
Kokka, A., Pulli, T., Honkavaara, E., Markelin, L., Kärhä, P., & Ikonen, E. (2019). Flat-field calibration method for hyperspectral frame cameras. *Metrologia*, *56*(5), 055001. https://doi.org/10.1088/1681-7575/ab3261
Kornprobst, P., Tumblin, J., & Durand, F. (2009). Bilateral Filtering: Theory and Applications. *Foundations and Trends in Computer Graphics and Vision*, *4*, 1-74. https://doi.org/10.1561/0600000020
Kramp, T., Kranenburg, R., & Lange, S. (2013). Introduction to the Internet of Things. In (pp. 1-10). https://doi.org/10.1007/978-3-642-40403-0_1
Krizhevsky, A., Sutskever, I., & Hinton, G. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *Neural Information Processing Systems*, *25*. https://doi.org/10.1145/3065386
Kumar, V., Sharma, K. V., Kedam, N., Patel, A., Kate, T. R., & Rathnayake, U. (2024). A comprehensive review on smart and sustainable agriculture using IoT technologies. *Smart Agricultural Technology*, *8*, 100487. https://doi.org/https://doi.org/10.1016/j.atech.2024.100487
Le, H., Le, T., Tran, S., Tran, H., & Thuy, N. (2012). Image Classification using Support Vector Machine and Artificial Neural Network. *International Journal of Information Technology and Computer Science*, *4*. https://doi.org/10.5815/ijitcs.2012.05.05
Li, C., Li, L., Jiang, H., Weng, K., Geng, Y., Li, L., Ke, Z., Li, Q., Cheng, M., Nie, W., Li, Y., Zhang, B., Liang, Y., Zhou, L., Xu, X., Chu, X., Wei, X., & Wei, X. (2022). *YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications*. https://doi.org/10.48550/arXiv.2209.02976
Li, J., Xu, W., Deng, L., Xiao, Y., Han, Z., & Zheng, H. (2023). Deep learning for visual recognition and detection of aquatic animals: A review. *Reviews in Aquaculture*, *15*(2), 409-433. https://doi.org/https://doi.org/10.1111/raq.12726
Li, P., Han, T., Ren, Y., Xu, P., & Yu, H. (2023). Improved YOLOv4-tiny based on attention mechanism for skin detection. *PeerJ Computer Science*, *9*, e1288. https://doi.org/10.7717/peerj-cs.1288
Lin, X., Ma, Y.-l., Ma, L.-z., & Zhang, R.-l. (2014). A survey for image resizing. *Journal of Zhejiang University SCIENCE C*, *15*(9), 697-716. https://doi.org/10.1631/jzus.C1400102
Liu, J., Qiao, W., & Xiong, Z. (2022). OAB-YOLOv5: One-Anchor-Based YOLOv5 for Rotated Object Detection in Remote Sensing Images. *Journal of Sensors*, *2022*, 8515510. https://doi.org/10.1155/2022/8515510
Liu, J., Yang, D., & Hu, F. (2022). Multiscale Object Detection in Remote Sensing Images Combined with Multi-Receptive-Field Features and Relation-Connected Attention. *Remote Sensing*, *14*, 427. https://doi.org/10.3390/rs14020427
Liu, M., Tang, L., Zhong, S., Luo, H., & Peng, J. (2021). Learning noise-decoupled affine models for extreme low-light image enhancement. *Neurocomputing*, *448*, 21-29. https://doi.org/https://doi.org/10.1016/j.neucom.2021.03.107
Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016, 2016//). SSD: Single Shot MultiBox Detector. Computer Vision – ECCV 2016, Cham.
Lowe, D. G. (1999, 20-27 Sept. 1999). Object recognition from local scale-invariant features. Proceedings of the Seventh IEEE International Conference on Computer Vision,
Machado, E., & Chen, L. (2019). *Visual Attention-Based Object Detection in Cluttered*

Environments. https://doi.org/10.1109/SmartWorld-UIC-ATC-SCALCOM-IOP-SCI.2019.00064

Maharana, K., Mondal, S., & Nemade, B. (2022). A review: Data pre-processing and data augmentation techniques. *Global Transitions Proceedings*, *3*(1), 91-99. https://doi.org/https://doi.org/10.1016/j.gltp.2022.04.020

Maraveas, C. (2023). Incorporating Artificial Intelligence Technology in Smart Greenhouses: Current State of the Art. *Applied Sciences*, *13*(1).

Maraveas, C., & Bartzanas, T. (2021). Application of Internet of Things (IoT) for Optimized Greenhouse Environments. *AgriEngineering*, *3*(4), 954-970.

Mascarenhas, V., & Keck, T. (2018, 2018//). Marine Optics and Ocean Color Remote Sensing. YOUMARES 8 – Oceans Across Boundaries: Learning from each other, Cham.

Mishra, S., Sarkar, U., Taraphder, S., Datta, S., Swain, D., Saikhom, R., Panda, S., & Laishram, M. (2017). Principal Component Analysis. *International Journal of Livestock Research*, 1. https://doi.org/10.5455/ijlr.20170415115235

Muksit, A. A., Hasan, F., Hasan Bhuiyan Emon, M. F., Haque, M. R., Anwary, A. R., & Shatabda, S. (2022). YOLO-Fish: A robust fish detection model to detect fish in realistic underwater environment. *Ecological Informatics*, *72*, 101847. https://doi.org/https://doi.org/10.1016/j.ecoinf.2022.101847

Mumuni, A., Fuseini. (2022). Data augmentation: A comprehensive survey of modern approaches. *Array*, *16*, 100258. https://doi.org/https://doi.org/10.1016/j.array.2022.100258

Muraina, I. (2022). *IDEAL DATASET SPLITTING RATIOS IN MACHINE LEARNING ALGORITHMS: GENERAL CONCERNS FOR DATA SCIENTISTS AND DATA ANALYSTS*.

Murat, A. A., & Kiran, M. S. (2025). A comprehensive review on YOLO versions for object detection. *Engineering Science and Technology, an International Journal*, *70*, 102161. https://doi.org/https://doi.org/10.1016/j.jestch.2025.102161

Murthy, C. B., Hashmi, M. F., Bokde, N. D., & Geem, Z. W. (2020). Investigations of Object Detection in Images/Videos Using Various Deep Learning Techniques and Embedded Platforms—A Comprehensive Review. *Applied Sciences*, *10*(9).

Mythili, M., Professor, A., & Kavitha, D. (2011). Efficient Technique for Color Image Noise Reduction. *T h e R e s e a r c h B u l l e t i n o f J o r d a n A C M , V o l . I I (I I I)*.

Nayan, A. A., Saha, J., Mahmud, K. R., Azad, A. K. A., & Kibria, M. G. (2020, 19-20 Dec. 2020). Detection of Objects from Noisy Images. 2020 2nd International Conference on Sustainable Technologies for Industry 4.0 (STI),

Nazaré, T., De Barros Paranhos da Costa, G., Contato, W., & Ponti, M. (2018). Deep Convolutional Neural Networks and Noisy Images. In (pp. 416-424). https://doi.org/10.1007/978-3-319-75193-1_50

Nedaie, A., & Najafi, A. A. (2018). Support vector machine with Dirichlet feature mapping. *Neural Networks*, *98*, 87-101. https://doi.org/https://doi.org/10.1016/j.neunet.2017.11.006

Nimitkul, S., Phromsuthirak, K., Taparhudee, W., Areekul, V., Oniam, V., & Arkronrat, W. (2022). Automated molting detection system for commercial soft-shell crab (Portunus pelagicus) production. *ScienceAsia*, *48*, 614. https://doi.org/10.2306/scienceasia1513-1874.2022.090

Nugrahaeni, R. A., & Mutijarsa, K. (2016, 5-6 Aug. 2016). Comparative analysis of machine learning KNN, SVM, and random forests algorithm for facial expression classification. 2016 International Seminar on Application for Technology of Information and Communication (ISemantic),

Owayjan, M., Dergham, A., Haber, G., Fakih, N., Hamoush, A., & Abdo, E. (2013). *Face Recognition Security System*.

Patil, A. (2021). IMAGE RECOGNITION USING MACHINE LEARNING. *International Journal of Engineering Applied Sciences and Technology*, *6*. https://doi.org/10.33564/IJEAST.2021.v06i01.027

Pawan, P., Manoj, G., Sumit, S., & Nagawat, A. (2010). Image De-noising by Various Filters for Different Noise. *International Journal of Computer Applications*, *9*. https://doi.org/10.5120/1370-1846

Pitakphongmetha, J., Suntiamorntut, W., & Charoenpanyasak, S. (2021). Internet of things for aquaculture in smart crab farming. Journal of Physics: Conference Series,

Prathusha, P., Jyothi, S., & Mamatha, D. M. (2018, 14-16 Feb. 2018). Enhanced Image Edge Detection Methods for Crab Species Identification. 2018 International Conference on Soft-computing and Network Security (ICSNS),

Rakhshanfar, M., & Amer, M. A. (2018, 7-10 Oct. 2018). Low-Frequency Image Noise Removal Using White Noise Filter. 2018 25th IEEE International Conference on Image Processing (ICIP),

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). *You Only Look Once: Unified, Real-Time Object Detection*. https://doi.org/10.1109/CVPR.2016.91

Redmon, J., & Farhadi, A. (2016). YOLO9000: Better, Faster, Stronger. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6517-6525.

Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement.

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*. https://doi.org/10.1109/TPAMI.2016.2577031

Rojek, L., Möller, M., Richter, M., Bischoff-Schaefer, M., & Hehl, K. (2023). PLANTSENS: A rail-based multi-sensor imaging system for redundant water stress detection in greenhouses. *Smart Agricultural Technology*, *4*, 100223. https://doi.org/https://doi.org/10.1016/j.atech.2023.100223

Shahriar, M., & Li, H. (2020). *A Study of Image Pre-processing for Faster Object Recognition*.

Shakoor, m. h., & Tajeripour, F. (2015). Circular Mean Filtering For Textures Noise Reduction. *Iranian Journal of Electrical and Electronic Engineering*, *11*, 1-9.

Shashoa, N. A., Ahmed, N., Jleta, I., & Abusaeeda, O. (2016). *Classification depend on linear discriminant analysis using desired outputs*. https://doi.org/10.1109/STA.2016.7952041

Shen, X., Wang, H., Li, Y., Gao, T., & Fu, X. (2024). Criss-cross global interaction-based selective attention in YOLO for underwater object detection. *Multimedia Tools and Applications*, *83*(7), 20003-20032. https://doi.org/10.1007/s11042-023-16311-y

Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, *6*(1), 60. https://doi.org/10.1186/s40537-019-0197-0

Sneha, & Kaul, A. (2022). Hyperspectral imaging and target detection algorithms: a review. *Multimedia Tools and Applications*, *81*(30), 44141-44206. https://doi.org/10.1007/s11042-022-13235-x

Son, S., & Jeong, Y. (2024). An Automated Fish-Feeding System Based on CNN and

GRU Neural Networks. *Sustainability*, *16*, 3675. https://doi.org/10.3390/su16093675

Soni, H., Sankhe, D., & Student. (2019). Image Restoration using Adaptive Median Filtering. *International Research Journal of Engineering IT & Scientific Research*, 2395-0056.

Srisha, R., & Khan, A. (2013). *Morphological Operations for Image Processing : Understanding and its Applications*.

Sun, Y., Li, Y., Li, S., Duan, Z., Ning, H., & Zhang, Y. (2023). PBA-YOLOv7: An Object Detection Method Based on an Improved YOLOv7 Network. *Applied Sciences*, *13*(18).

Tang, C., Hu, H., Wei, P., Duan, Z., Zhang, G., & Qian, Y. (2020). An improved YOLOv3 algorithm to detect molting in swimming crabs against a complex background. *Aquacultural Engineering*, *91*, 102115. https://doi.org/10.1016/j.aquaeng.2020.102115

Tang, C., Wang, D., Tan, A.-H., & Miao, C. (2017). *EEG-Based Emotion Recognition via Fast and Robust Feature Smoothing*. https://doi.org/10.1007/978-3-319-70772-3_8

Taunk, K., De, S., Verma, S., & Swetapadma, A. (2019). *A Brief Review of Nearest Neighbor Algorithm for Learning and Classification*. https://doi.org/10.1109/ICCS45141.2019.9065747

Tian, H., Wang, T., Liu, Y., Qiao, X., & Li, Y. (2020). Computer vision technology in agricultural automation —A review. *Information Processing in Agriculture*, *7*(1), 1-19. https://doi.org/https://doi.org/10.1016/j.inpa.2019.09.006

Ullah, I., Adhikari, D., Su, X., Palmieri, F., Wu, C., & Choi, C. (2025). Integration of data science with the intelligent IoT (IIoT): Current challenges and future perspectives. *Digital Communications and Networks*, *11*(2), 280-298. https://doi.org/https://doi.org/10.1016/j.dcan.2024.02.007

van der Maaten, L., & Hinton, G. (2008). Viualizing data using t-SNE. *Journal of Machine Learning Research*, *9*, 2579-2605.

van der Maaten, L., Postma, E., & Herik, H. (2007). Dimensionality Reduction: A Comparative Review. *Journal of Machine Learning Research - JMLR*, *10*.

Verma, R., & Ali, J. (2013). A comparative study of various types of image noise and efficient noise removal techniques. *International Journal of Advanced Research in Computer Science and Software Engineering*, *3*, 617-622.

Villar, S., Torcida, S., & Acosta, G. (2017). Median Filtering: A New Insight. *Journal of Mathematical Imaging and Vision*, *58*, 1-17. https://doi.org/10.1007/s10851-016-0694-0

Viola, P., & Jones, M. (2001, 8-14 Dec. 2001). Rapid object detection using a boosted cascade of simple features. Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001,

Waiho, K., Ikhwanuddin, M., Baylon, J. C., Jalilah, M., Rukminasari, N., Fujaya, Y., & Fazhan, H. (2021). Moult induction methods in soft-shell crab production. *AQUACULTURE RESEARCH*, *52*(9), 4026-4042. https://doi.org/10.1111/are.15274

Wang, C.-Y., Bochkovskiy, A., & Liao, H.-y. (2022). *YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors*. https://doi.org/10.48550/arXiv.2207.02696

Wang, C. Y., Bochkovskiy, A., & Liao, H. Y. (2022). *YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors*. https://doi.org/10.48550/arXiv.2207.02696

Wang, P., Fan, E., & Wang, P. (2021). Comparative analysis of image classification algorithms based on traditional machine learning and deep learning. *Pattern Recognition Letters*, *141*, 61-67. https://doi.org/https://doi.org/10.1016/j.patrec.2020.07.042

Wang, W., & Wang, Y. (2021). Underwater target detection system based on YOLO v4. 2021 2nd International Conference on Artificial Intelligence and Information Systems,

Wen, Q., Li, X., Wan, H., Huang, J., Chen, L., Li, D., & Luo, T. (2024). Flat-field methods for color image sensors. *Journal of the Optical Society of America B*, *41*, 2748-2755. https://doi.org/10.1364/JOSAB.529622

Xu, S., Zhang, M., Song, W., Mei, H., He, Q., & Liotta, A. (2023). A systematic review and analysis of deep learning-based underwater object detection. *Neurocomputing*, *527*, 204-232. https://doi.org/https://doi.org/10.1016/j.neucom.2023.01.056

Yang, S., Lin, C., Liao, K., Zhao, Y., & Liu, M. (2020). Unsupervised fisheye image correction through bidirectional loss with geometric prior. *Journal of Visual Communication and Image Representation*, *66*, 102692. https://doi.org/https://doi.org/10.1016/j.jvcir.2019.102692

Yaroslavsky, L. (2004). Nonlinear Filters in Image Processing. In (pp. 509-540). https://doi.org/10.1007/978-1-4757-4988-5_12

Yu, J., Zheng, H., Xie, L., Zhang, L., Yu, M., & Han, J. (2023). Enhanced YOLOv7 integrated with small target enhancement for rapid detection of objects on water surfaces [Original Research]. *Frontiers in Neurorobotics*, *17*. https://doi.org/10.3389/fnbot.2023.1315251

Yun, S., Han, D., Chun, S., Oh, S. J., Yoo, Y., & Choe, J. (2019). *CutMix: Regularization Strategy to Train Strong Classifiers With Localizable Features*. https://doi.org/10.1109/ICCV.2019.00612

Zhang, M., Xu, S., Song, W., He, Q., & Wei, Q. (2021). Lightweight Underwater Object Detection Based on YOLO v4 and Multi-Scale Attentional Feature Fusion. *Remote Sensing*, *13*, 4706. https://doi.org/10.3390/rs13224706

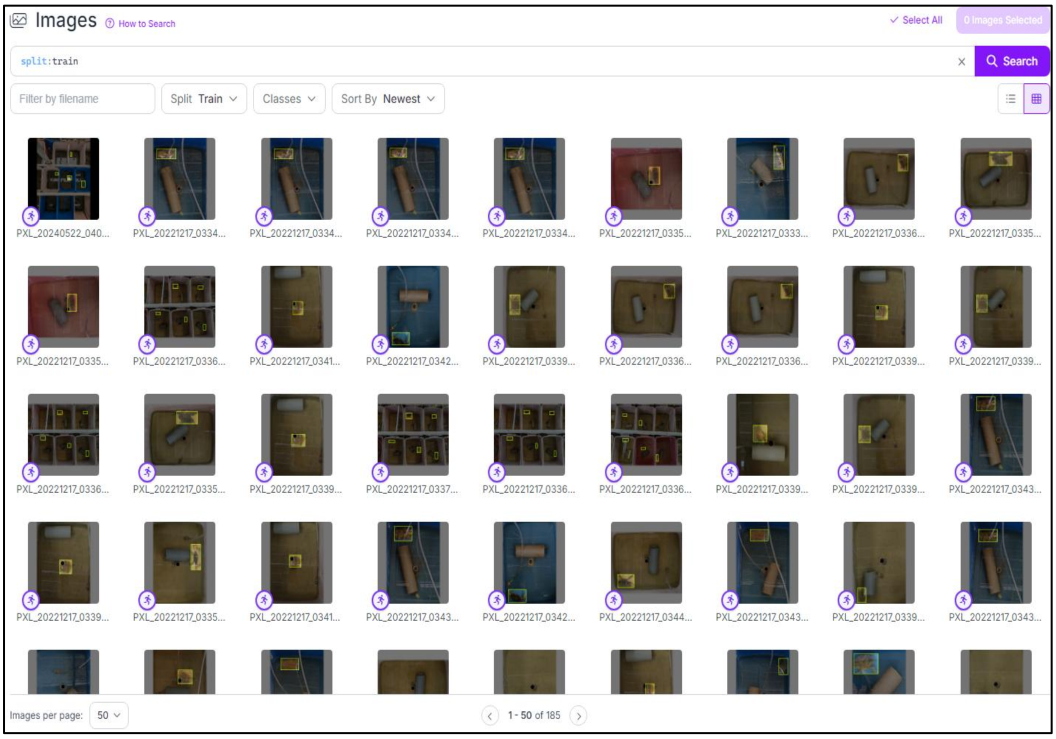
Zuo, W., Zhang, K., & Zhang, L. (2018). Convolutional Neural Networks for Image Denoising and Restoration: Fundamentals, Open Challenges and New Trends. In (pp. 93-123). https://doi.org/10.1007/978-3-319-96029-6_4

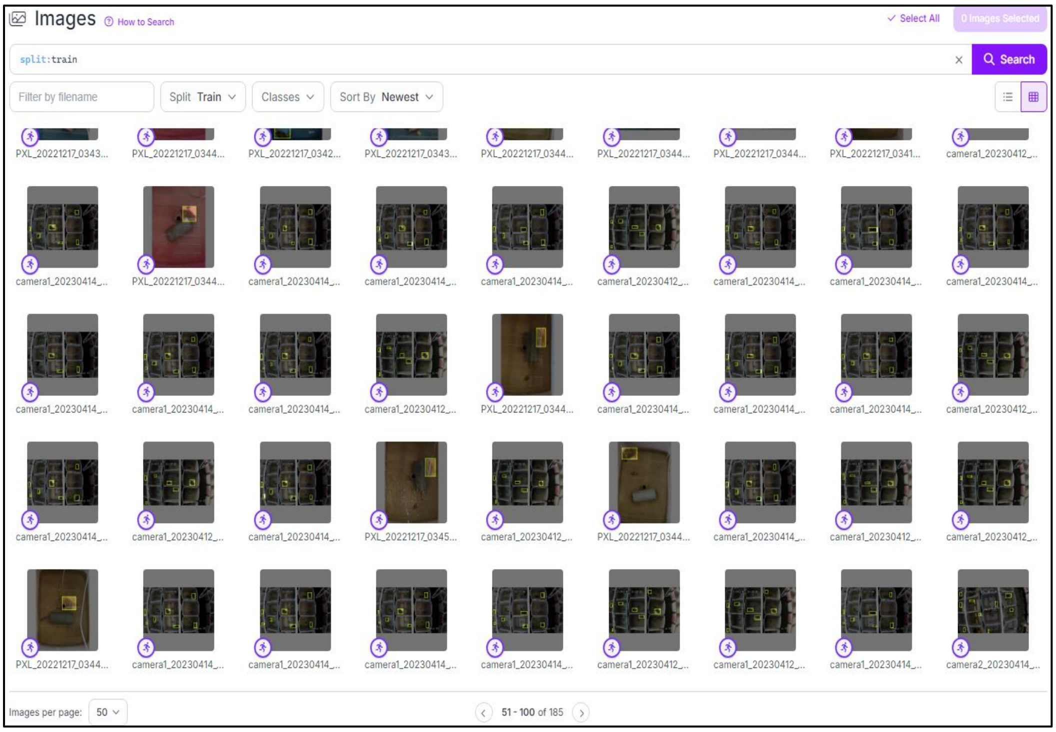
APPENDICES

APPENDIX A
Training progress for YOLOv7

```
     Epoch   gpu_mem       box       obj       cls     total    labels  img_size
     95/99     5.77G   0.03376  0.003665         0   0.03742         9       640: 100%|█| 7/7 [00:09<00:00,  1.30s/it]
               Class    Images    Labels         P         R    mAP@.5  mAP@.5:.95: 100%|█| 1/1 [00:00<00:00,
                 all        12        12     0.857         1     0.942     0.585

     Epoch   gpu_mem       box       obj       cls     total    labels  img_size
     96/99     5.77G   0.03406  0.003923         0   0.03798        12       640: 100%|█| 7/7 [00:09<00:00,  1.31s/it]
               Class    Images    Labels         P         R    mAP@.5  mAP@.5:.95: 100%|█| 1/1 [00:00<00:00,
                 all        12        12     0.857         1     0.942      0.61

     Epoch   gpu_mem       box       obj       cls     total    labels  img_size
     97/99     5.77G   0.03049  0.003406         0   0.03389         5       640: 100%|█| 7/7 [00:09<00:00,  1.30s/it]
               Class    Images    Labels         P         R    mAP@.5  mAP@.5:.95: 100%|█| 1/1 [00:00<00:00,
                 all        12        12     0.857         1     0.942     0.637

     Epoch   gpu_mem       box       obj       cls     total    labels  img_size
     98/99     5.77G   0.03078  0.003311         0   0.03409         5       640: 100%|█| 7/7 [00:09<00:00,  1.30s/it]
               Class    Images    Labels         P         R    mAP@.5  mAP@.5:.95: 100%|█| 1/1 [00:00<00:00,
                 all        12        12     0.857         1     0.942     0.618

     Epoch   gpu_mem       box       obj       cls     total    labels  img_size
     99/99     5.77G   0.03345  0.003868         0   0.03731        13       640: 100%|█| 7/7 [00:09<00:00,  1.30s/it]
               Class    Images    Labels         P         R    mAP@.5  mAP@.5:.95: 100%|█| 1/1 [00:00<00:00,
                 all        12        12     0.923         1      0.95     0.648
100 epochs completed in 0.326 hours.
```

APPENDIX B
Example of training image data split using Roboflow





APPENDIX C

Example of validation image data split using Roboflow

APPENDIX D
Example of testing image data split using Roboflow

AUTHOR'S PROFILE

Mohammad Affaiq Bin Aini obtained Diploma in Industrial Chemistry in 2019, Bachelor of Science in Chemistry with Management (Hons.) in 2022, and Master of Science in Information Technology in 2026 from Universiti Teknologi MARA. His Master thesis title is Image Detection for Monitoring Soft Shell Crab Using Internet of Things. His work applies the use of Computer Vision, Data Augmentation, Preprocessing, Classification and Object Detection. He was a research assistant for SDEC funded project from 2022-2024.

LIST OF PUBLICATION

Aini, M. a. B., Leong, S. H., Yong, Y. T., Lee, B. Y., Zhao, X., Manaf, S. R., Abdullah, F., & Khong, H. Y. (2025). Image enhancement for detection of underwater moulted crabs in greenhouse soft-shell crab farming using deep learning. *Smart Agricultural Technology*, *12*, 101223. https://doi.org/10.1016/j.atech.2025.101223